

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ

ХАРЬКОВСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ
РАДИОЭЛЕКТРОНИКИ

ISSN 0555-2656

БИОНИКА
ИНТЕЛЛЕКТА
ИНФОРМАЦИЯ, ЯЗЫК, ИНТЕЛЛЕКТ

№ 1 (72)

2010

НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Основан в 1967 г.

Свидетельство о государственной регистрации КВ № 12072-943 ПР от 07.12.2006

Журнал включен в список специальных изданий ВАК Украины
по техническим наукам
(приложение к постановлению ВАК Украины № 1-05/7 от 04.07.2006)

СОДЕРЖАНИЕ

<i>Четвериков Г.Г.</i> Концепція уніфікації методів та засобів побудови просторових багатозначних структур мовних систем	3
<i>Пуятин Е.П., Оробинский П.А.</i> Построение 12-факторной модели человеческого лица	12
<i>Гришко А.А., Удовенко С.Г., Чалая Л.Э.</i> Определение стратегий в трейдинговых системах на основе методов машинного обучения	18
<i>Левыкин В.М., Чальи С.Ф., Кривчикова А.А., Селезнев С.Н.</i> Логическая модель представления бизнес-процесса для решения задач интеллектуального анализа процессов	24
<i>Баранов О.В., Грицай Д.В., Гребенник І.В., Панкратов О.В., Чугай А.М.</i> Розміщення прямокутних графічних елементів в поліграфічному виробництві	29
<i>Ведмедь А.Г., Машталир В.П., Сакало Е.С.</i> Нейронная сеть и алгоритм ее обучения для анализа независимых компонент в задачах обработки изображений	33
<i>Субботин С.А.</i> Критерии индивидуальной информативности и методы отбора экземпляров для построения диагностических и распознающих моделей	38
<i>Дубницкий В.Ю., Кобылин А.М., Кобылин О.А.</i> Интервальное оценивание параметров для определения надёжности программного обеспечения	43
<i>Бескоровайный В.В., Соболева Е.В.</i> Идентификация частной полезности многофакторных альтернатив с помощью S-образных функций	50
<i>Авраменко В.П., Ткаченко В.П., Чібірев А.Д.</i> Методи підвищення якості захисту інформації на основі використання фрактальних функцій	55
<i>Крючковский В.В., Писклакова О.А., Филиппская Д.И.</i> Модель задачи распределения ресурсов в условиях многокритериальности и неопределенности	61
<i>Машталир С.В., Постульга С.В., Щербинин К.С.</i> Поиск изменения сцен в видеоданных на базе анализа результатов сегментации	65
<i>Кораблев Н.М., Фомичев А.А.</i> Исследование иммунных операторов в задаче кластеризации объектов	70
<i>Крыгин М.Ю.</i> Текст на естественном языке как объект статистического анализа	75
<i>Вечірська І.Д., Федорова Т.М., Четвериков Г.Г.</i> Побудова функціональних моделей елементів комп'ютерної лінгвістики	83
<i>Любченко В.А., Кобылин О.А.</i> Использование уровней вейвлет-преобразования для выделения контура изображения	89
<i>Русакова Н.Е.</i> Модель устной речи	94
<i>Пелешко Д.Д., Рашкевич Ю.М.</i> Задача факторизации пространств, определенных на топологиях изображений	98
<i>Збитнева М.В.</i> Интеллектуальные агенты для создания псевдо контента веб-сайтов	104
<i>Виклюк Я.І.</i> Методи фрактальної геометрії та нечіткої логіки, як основа математичного моделювання просторової організації туристичних поселень	109
<i>Танянский С.С., Мальков Ю.А.</i> Организация запросов к распределенным данным средствами логического программирования	118
<i>Стопченко Г.И., Макрушан И.А., Билан С.В.</i> Задачи и концепции методов принятия многокритериальных решений в интеллектуальных системах	122
<i>Воскобойникова А.А.</i> Использование методов планирования для решения задач в информационных системах	126
Об авторах	130
Правила оформления рукописей для авторов научно-технического журнала «Бионика интеллекта»	133

УДК 007. 519.714:519.766



КОНЦЕПЦІЯ УНІФІКАЦІЇ МЕТОДІВ ТА ЗАСОБІВ ПОБУДОВИ ПРОСТОРОВИХ БАГАТОЗНАЧНИХ СТРУКТУР МОВНИХ СИСТЕМ

Г.Г. Четвериков

Харківський національний університет радіоелектроніки, м. Харків, chetvergg@kture.kharkov.ua

Стаття присвячена аналізу проблеми створення мовних систем штучного інтелекту. Проведено аналіз алгебро-логічної структури природної мови, що дозволяє моделювати її елементи на фонетичному та морфологічному рівнях на підставі уніфікації логічного та апаратного підходів.

АЛГЕБРА СКІНЧЕННИХ ПРЕДИКАТИВ, ЛОГІКА, ПРИРОДНА МОВА, ШТУЧНИЙ ІНТЕЛЕКТ,
k-ЗНАЧНА СТРУКТУРА, АСП-СТРУКТУРА

Вступ

В умовах роботи реальних систем із високим рівнем невизначеності задач (середовищ) та інформації для побудови інтелектуальних систем (ІС) неминуче використання нових інформаційних технологій, орієнтованих на потоки контекстно залежної інформації. Фактично необхідною є розробка саме природномовних принципів побудови інтелектуального інтерфейсу. Очевидно, що єдиним, відомим нам, об'єктивним носієм моралі і інтелекту є людина, а виразником, засобом до зовнішнього спілкування, є людська мова. Отже, єдиним можливим шляхом найбільш об'єктивного, повного і ефективного створення інтелектуальних систем та систем штучного інтелекту (ШтІ) є шлях аналізу, моделювання та синтезу мовного інтелектуального інтерфейсу [1,2].

На сьогоднішній день таким середовищем, можливо єдиним, є інтелектуальні системи, у яких, починаючи з рівня застосування природної мови, в повній мірі постає питання про необхідність застосування *k*-значної логіки, структур і кодування. Визначивши середовище, якому необхідні багатозначні структури (БС), легше визначити, які властивості та характеристиками вони повинні мати і за якими принципами бути побудовані. З іншого боку, бурхливий розвиток комп'ютеризації, проникнення обчислювальної техніки в усі сфери науки, промисловості, суспільного життя, її використання для розв'язання найскладніших задач сьогодення вступили у протиріччя з технологією обробки даних традиційними фон-нейманівськими комп'ютерами. Спостерігається криза її основ, пов'язана з специфікою архітектури та принципів дії фон-нейманівського процесора з застосуванням послідовних алгоритмів роботи та виключно двозначних елементів, структур і кодування. Перелічені проблеми вимагають вирішення задач розвитку нових принципів організації елементів, структур і методів кодування для обчислювальних систем, зокрема систем ШтІ [3,4].

Людський мозок зберігає об'єктивну модель світової реальності, інакше людина не змогла б

еволюціонувати та вдосконалюватись у процесі свого розвитку. Таким чином можна зробити висновки про доцільність моделювання принципів дії людського мозку для задач створення ШтІ. Оскільки усі розумові здібності, що необхідно передати машині із ШтІ, вже наявні в людини на достатньо високому рівні розвитку, і ніякий інший інтелект, крім людського, науці недоступний, то міркувати машини повинні за тими ж законами, що й людина. Нейрофізіологічні дослідження принципів дії природного інтелекту мозку людини виявили у ньому наявність дво- та багатозначного (*k*-значного) кодування, просторового характеру активності мереж нервових клітин і організації діяльності мозку.

Звідси випливають основні вимоги щодо властивостей базових елементів і структур для побудови новітніх високоефективних систем ШтІ. Вони повинні реалізувати функції багатозначної логіки та кодування, володіти властивостями універсальності, просторовості, гібридності, гнучкого переналагодження без зміни структури, ієрархічності, за складністю порівнюватись із складністю задач, що розв'язуються. Найближчими за вказаними властивостями є багатозначні універсальні просторові елементи та структури [2,4,5].

Можливо прогнозувати і подальший розвиток інтелектуальних систем. Це можна зробити виходячи всього з двох достатньо очевидних міркувань. По-перше, інтелектуальні системи повинні одержати свою власну "інформаційну машину" – машину, що пододала теоретичні обмеження кінцевих автоматів і практично нереалізованої машини Тюрінга. Тут необхідне наукове і технічне рішення, що долає бар'єр "необчислюваності". По-друге, сам факт побудови такої машини, практика спілкування з нею приведуть до остаточного переосмислення поняття і значення інформаційної науки – інформатики. Достатньо очевидний факт неможливості існування інформаційної машини такого роду інакше, як у безперервному режимі засвоєння і реструктуризації інформації, приведе до розуміння інтелектуальної системи як системи досить динамічної.

Отже, основні постулати теорії інтелектуальних систем [1, 4]:

- комп'ютер фон-нейманівської архітектури, кінцевий автомат за своєю суттю, не може бути інструментом створення інтелектуальних систем;
- підвищується інтелектуальність системи при зниженні її точності і навпаки;
- теорія повинна розглядати можливість побудови абстрактних конструкцій, що реалізують не обчислювані в звичному значенні об'єкти;
- необхідна розробка природномовних принципів побудови інтелектуальних систем та систем ШтІ.

На даний час проблеми у розвитку систем ШтІ ставлять питання про застосування орієнтації на досягнення високорівневої технології обробки інформації (тобто отримання нової якості), що виявляється у намаганнях реалізувати на фон-неймановських комп'ютерах системи ШтІ. Отже, сучасні, нові вимоги до технології обробки інформації можна звести до двох моментів: необхідність вирішення проблем, що погано формалізуються, та наявність кінцевого користувача, який за своєю професійною діяльністю не є програмістом [2,3].

Таким чином, приходимо до розуміння одного з варіантів шляхів створення систем ШтІ – це шлях аналізу, моделювання та синтезу мовного інтелектуального інтерфейсу за допомогою засобів k -значної логічної системи (зокрема алгебри скінченних предикатів (АСП)) та k -значних структур і кодування [2, 6-8].

1. Мета роботи

Основним завданням цієї роботи є викладення новоствореної концепції уніфікації методів та засобів побудови просторових багатозначних структур мовних систем. Предмет досліджень – апаратні методи розв'язання рівнянь алгебри предикатів та засоби побудови універсальних багатовходових k -значних просторових структур мовних систем штучного інтелекту. Методи дослідження ґрунтуються на використанні теорії моделювання та базуються на математичному апараті теорії інтелекту, зокрема АСП, для отримання формального опису системи фонетичних ознак, букво-фонемних відношень та морфології української мови; при синтезі швидкодіючих багатозначних структур мовних систем використано апарат теорії інформації, множин, інтелекту, кодування та лексикографічних систем і лексикографічних числень [9 – 11].

2. Вибір математичного апарату

У роботі [10] для задач моделювання режимів автоматизованої обробки мови розроблено та проаналізовано мінімальну систему аксіом АСП. Доведено: АСП є повною алгеброю у тому розумінні, що за допомогою її формул можна в аналітичному

вигляді записати довільний скінченний предикат. Задача інтелектуалізації та українізації цифрових мереж і структур не передбачає простого й швидкого вирішення. Головна перешкода, на яку наштовхуються всі спроби радикально підвищити інтелектуальні здібності ЕОМ, полягає у недосконалості машинних мов. Вони значно поступаються природним мовам. Збільшується кількість прихильників того, що створення «машинного інтелекту» вимагає вивчення й моделювання людського інтелекту та, у першу чергу, природної мови, що лежить у його основі. Стає все очевиднішим, що при подальшому вдосконаленні машинних мов необхідно ширше використовувати організацію людської мови. Механізм природної мови, не дивлячись на поверхневу легкість користування людиною, надзвичайно складний й до того ж мало вивчений. Отже, розробка систем обробки мовної інформації часто базується на недостатньо міцному лінгвістичному фундаменті. Багато хто з дослідників, що займалися автоматизацією мовної діяльності, перейшли до поглибленого вивчення, дослідження й моделювання окремих сторін механізму природної мови.

Наявність АСП відкриває можливість переходу від алгоритмічного опису інформаційних процесів до опису їх у вигляді рівнянь. А рівняння в свою чергу задають відношення між змінними. Усі змінні в рівнянні рівноправні, будь-які з них можуть виступати як у ролі незалежних, так і в ролі залежних. При цьому перевагою рівнянь перед алгоритмами є те, що можна розрахувати реакцію системи навіть при неповній визначеності вхідних сигналів, у той час як не повністю розроблений алгоритм є непрацездатним. Встановлено, що за умов зміни знань про об'єкт система рівнянь АСП, покладених на структуру системи, завжди готова до використання, а алгоритм часто вимагає докорінної зміни її структури. За допомогою формул АСП будуюмо АСП-структури, які реалізують відповідні скінченні предикати. Це є аналог побудови комбінаційних схем за формулами алгебри логіки. В залежності від рівня функціонально-структурної реалізації маємо АСП-структури першого, другого та третього роду.

3. Природномовні принципи побудови систем штучного інтелекту

Виходячи із поставленої мети та враховуючи основні аксіоми теорії інтелектуальних систем управління, інтегруємо необхідні та уже розроблені природномовні принципи побудови інтелектуального інтерфейсу і систем ШтІ. У цій роботі, перш за все, хотілося б показати необхідність і можливість розробки загальної теорії побудови інтелектуального інтерфейсу та систем штучного інтелекту, яка могла б стати методологічною основою цілеспрямованого створення нових інформаційних та інтелектуальних технологій.

4. Алгебро-логічні засоби моделювання природної мови

Розглянуто алгебро-логічні засоби моделювання природної мови, аналіз та дослідження її алгебро-логічної структури. Природна мова (ПМ), як явище інтелектуальної діяльності людини, є дуже складним об'єктом. Але маючи формальний опис природної мови, його можна реалізувати на ЕОМ і таким чином наділити машину здібністю володіти природною мовою. Необхідно, щоб машина пристосовувалась до людини та розмовляла її мовою, а не навпаки — людина прилаштовувалась до машини. Природна мова для людини найзручніша. Людина ніколи не проміняє її ні на мову математики, ні на мову програмування. Адже немає іншої галузі знань, яка б в більшій мірі сприяла підвищенню темпів комп'ютеризації та інформатизації суспільства. Тоді можна буде наділити машину здібністю до природної мови. Якщо з'явиться можливість виявити з природної мови шляхом її аналізу більш досконалу алгебраїчну мову, це дуже збільшить можливості розробників інформаційних систем та нових інформаційних технологій. Такий алгебро-логічний апарат дозволить розширити можливості розробників, які займаються створенням нових інформаційних технологій. Отже, концептуально-методологічний підхід до ПМ (з математичної точки зору) дозволяє сприймати її як деяку алгебру, а тексти — як формули цієї алгебри.

Таким чином, розглянемо перспективи побудови ЕОМ за принципами роботи мозку. Відправною точкою у цих дослідженнях буде аналогія між багатозначними структурами з просторовими принципами побудови та мікроструктурою кори мозку.

Під час опису лінгвістичних зв'язків природної мови використовується метод зрізу як основний методологічний спосіб [10]. Згідно з цим методом фрагмент тексту X , який нас цікавить, відокремлюють в оточуючому контексті, а самим контекстом нехтують. Дію контексту замінюють набором ознак Y . Потім вивчається і математично описується лінгвістичне відношення $L: X L Y$, що зв'язує фрагмент тексту X з набором його ознак Y . Метод зрізу дозволяє розбити складну задачу математичного опису лінгвістичних закономірностей на множину простіших задач без неприпустимого спрощення й огрублення загальної задачі.

Повний математичний опис фонетичних й морфологічних зв'язків можна досягти простим об'єднанням їх у єдину систему рівнянь, що описують лінгвістичні відношення. У цій системі ознаки відіграють тільки другорядну роль: вони використовуються як проміжні змінні.

Вибір кожної ознаки й області її значень можна здійснити довільним чином. Але довільний набір ознак не можна признати оптимальним. Критерієм оптимальності служить складність отриманої сис-

теми рівнянь. Проте складність системи рівнянь не завжди легко оцінити. Саме цим пояснюються труднощі, що виникають у лінгвістів під час вирішення питання вибору оптимальної серед можливих систем ознак. Викладений нижче підхід досить повно визначає теоретичний статус поняття ознаки.

5. Фонетичний аналіз

У цьому розділі об'єктом моделювання є відношення, що базуються на лінгвістичних зв'язках, які існують між різними елементами фонетичного рівня української мови. На підставі вивчення даних фонетики, математично описуються відношення, що зв'язують окремі фонемні з системою їх фонетичних ознак:

y_1 — ознака голосності звуку зі значеннями: **г** — голосна, **п** — приголосна;

y_2 — ознака шумності приголосних зі значеннями: **ш** — шумова, **нш** — нешумова;

y_3 — ознака вокалізації звуку зі значеннями: **дз** — дзвінка, **г** — глуха;

y_4 — ознака проривності звуку зі значеннями: **пр** — проривна, **щ** — щілинна;

y_5, y_6 — ознаки веляризації та лабілізації звуку зі значеннями: **м, с, в** — мала, середня, велика відповідно;

y_7 — ознака палаталізації зі значеннями: **т, м** — тверда, **м'** — м'яка відповідно;

y_8 — ознака назалізації зі значеннями: **р** — ротова, **н** — носова;

y_9 — ознака вібрантності зі значеннями: **с** — спокійна, **т** — тремтяча;

y_{10} — ознака місця артикуляції звуку зі значеннями: **г** — губна, **п** — передньоязикова, **з** — задньоязикова;

y_{11} — ознака локалізації передньоязикової шумової фонемі зі значеннями: **з** — зубна, **п** — піднебінна;

y_{12} — ознака африкативності передньоязикових шумових фонем зі значеннями: **аф** — африката, **неаф** — неафриката.

Встановлено зв'язки між звуками Y мови у визначеному фонетичному контексті Z та відповідними їм буквами українського алфавіту X у вигляді тернарного відношення $\Phi(X, Y, Z)$, яке названо буквофонемним.

Математична модель лінгвістичних зв'язків, що регулюють процес переходу від фонетичного подання слів до їх графічного запису й навпаки, матиме вигляд:

$$\Phi = \Phi_1 \& \Phi_2 \& \dots \& \Phi_r,$$

де $\Phi_1 = 1; \Phi_2 = 1, \dots, \Phi_r = 1$ — різні, незалежні одне від одного зв'язки між X, Y, Z , що фактично існують в українській мові; r — загальна їх кількість.

Під час переходу від значень фонетичних ознак до фонетичного зображення звуку формується

фонема, тобто знак, що позначає клас звуків, які тяжіють до певної букви тексту. Зворотне перетворення полягає в заміні фонетичного знаку набором відповідних йому значень фонетичних ознак звуку.

Фонема як знак: $\text{ФОН}(X) = x^A \vee x^B \vee x^V \vee x^Г \vee x^Г \vee x^Д \vee x^E \vee x^E \vee x^Ж \vee x^З \vee x^И \vee x^I \vee x^I \vee x^Й \vee x^K \vee x^Л \vee x^М \vee x^Н \vee x^О \vee x^П \vee x^Р \vee x^С \vee x^Т \vee x^У \vee x^Ф \vee x^Х \vee x^Ц \vee x^Ч \vee x^Ш \vee x^Щ \vee x^Ю \vee x^Я$.

Ознака голосності звуку y_1 :

$\text{Голосн}(X) = x^A \vee x^E \vee x^I \vee x^O \vee x^Y \vee x^И$;

$\text{Приголсн}(X) = x^B \vee x^B \vee x^Г \vee x^Г \vee x^Д \vee x^E \vee x^Ж \vee x^З \vee x^Й \vee x^K \vee x^Л \vee x^М \vee x^Н \vee x^П \vee x^Р \vee x^С \vee x^Т \vee x^Ф \vee x^Х \vee x^Ц \vee x^Ч \vee x^Ш \vee x^Щ$.

Ознака шумності звуку y_2 :

$\text{Шум}(X) = x^B \vee x^B \vee x^Г \vee x^Д \vee x^Ж \vee x^З \vee x^K \vee x^С \vee x^П \vee x^Т \vee x^Ф \vee x^Х \vee x^Ц \vee x^Ч \vee x^Ш$;

$\text{Сон}(X) = x^Л \vee x^М \vee x^Н \vee x^Р \vee x^Й$;

$y_1 \text{ ПРИГ} \rightarrow y_2 \text{ НШУМ} \wedge \text{Сон}(X) \vee y_2 \text{ ШУМ} \wedge \text{Шум}(X) = 1$.

Ознака вокалізації звуку y_3 :

$\text{Глух}(X) = x^K \vee x^П \vee x^С \vee x^Т \vee x^Ф \vee x^Х \vee x^Ц \vee x^Ч \vee x^Ш$;

$\text{Дзвн}(X) = x^B \vee x^B \vee x^Г \vee x^Д \vee x^Ж \vee x^З \vee x^Ц \vee x^Ч$. $y_1 \text{ ПРИГ} \rightarrow y_3 \text{ ДЗ} \wedge \text{Дзвн}(X) \vee y_3 \text{ ГЛУХ} \wedge \text{Глух}(X) = 1$.

Ознака проривності звуку y_4 :

$\text{Прор}(X) = x^B \vee x^Г \vee x^Д \vee x^K \vee x^П \vee x^Т \vee x^Х \vee x^Ц \vee x^Ч$;

$\text{Щіл}(X) = x^В \vee x^Г \vee x^Ж \vee x^З \vee x^С \vee x^Ф \vee x^Ш \vee x^Щ$;

$y_1 \text{ ПРИГ} \wedge y_2 \text{ ШУМ} \rightarrow y_4 \text{ ПРОР} \wedge \text{Прор}(X) \vee y_4 \text{ ЩІЛ} \wedge \text{Щіл}(X) = 1$.

Ознака веляризації звуку y_5 :

$\text{Мвл}(X) = x^A \vee x^I \vee x^Я$; $\text{Свл}(X) = x^E \vee x^E \vee x^O$;

$\text{Ввл}(X) = x^I \vee x^Y \vee x^И \vee x^Ю$;

$z^1 \rightarrow y_5^M \wedge \text{Мвл}(X) \vee y_5^C \wedge \text{Свл}(X) \vee y_5^E \wedge \text{Ввл}(X) = 1$.

Ознака лабілізації звуку y_6 :

$\text{Млб}(X) = x^A \vee x^E \vee x^I \vee x^И \vee x^E \vee x^Я$; $\text{Слб}(X) = x^E \vee x^O$;

$\text{Влб}(X) = x^Y \vee x^Ю$; $z^1 \rightarrow y_6^M \wedge \text{Млб}(X) \vee y_6^C \wedge \text{Слб}(X) \vee y_6^B \wedge \text{Влб}(X) = 1$.

Ознака палаталізації звуку y_7 :

$\text{Мгол}(X) = x^E \vee x^I \vee x^I \vee x^Ю \vee x^Я$;

$\text{Тгол}(X) = x^A \vee x^O \vee x^Y \vee x^И \vee x^E$; $z^1 \rightarrow y_7^T \wedge \text{Тгол}(X) \vee y_7^M \wedge \text{Мгол}(X) = 1$.

Рівняння зв'язку фонем X з ознаками y_5, y_6, y_7 :

$z^1 \rightarrow y_5^M (y_6^M (y_7^T x^A \vee y_7^M x^Я) \vee y_6^B \wedge (y_7^T x^И \vee y_7^M x^I)) \vee y_5^C ((y_7^T x^O \vee y_7^M x^I) \vee (y_7^T x^E \vee y_7^M x^E)) \vee y_5^B (y_7^T x^Y \vee y_7^M x^Ю) = 1$.

Ознака назалізації звуку y_8 :

$\text{Рсн}(X) = x^Й \vee x^Л \vee x^М$; $\text{Нсн}(X) = x^М \vee x^Н$;

$z^2 \rightarrow y_8^P \wedge \text{Рсн}(X) \vee y_8^H \vee \text{Нсн}(X) = 1$.

Ознака вібрантності звуку

y_9 : $\text{Ссн}(X) = x^Й \vee x^Л \vee x^М \vee x^Н \vee x^В$;

$z^2 \rightarrow y_9^C \wedge \text{Ссн}(X) \vee y_9^T \wedge x^T = 1$

Ознака місця артикуляції звуку

y_{10} : $\text{Псн}(X) = x^Л \vee x^Н \vee x^Р$;

$z^2 \rightarrow y_{10}^Г \wedge x^М \vee y_{10}^П \wedge \text{Псн}(X) \vee y_{10}^З \wedge x^Й = 1$.

Рівняння зв'язку фонем X з ознаками y_8, y_9, y_{10} :

$z^2 \rightarrow y_{10}^З x^Й \vee y_8^P y_{10}^П x^Л \vee y_8^H (y_{10}^Г x^М \vee y_{10}^П x^Н) \vee y_9^T x^P = 1$.

Індивідуальні предикати для передньоязикових, задньоязикових та губних шумових фонем:

$\text{Пшф}(X) = x^Д \vee x^Ж \vee x^З \vee x^ДЖ \vee x^ДЗ \vee x^С \vee x^Т \vee x^Ц \vee x^Ч \vee x^Ш$;

$\text{Зшф}(X) = x^Г \vee x^K \vee x^Х$; $\text{Гшф}(X) = x^Б \vee x^В \vee x^Г \vee x^Г \vee x^Ф$;

$y_2 \text{ Ш} \rightarrow y_{10}^Г \text{Гшф}(X) \vee y_{10}^П \text{Пшф}(X) \vee y_{10}^З \text{Зшф}(X) = 1$.

Індивідуальні предикати для зубних та піднебінних передньоязикових шумових фонем:

$\text{Зпш}(X) = x^Д \vee x^З \vee x^ДЖ \vee x^ДЗ \vee x^С \vee x^М \vee x^Ц$;

$\text{Ппш}(X) = x^Д \vee x^Ж \vee x^Т \vee x^Ч \vee x^Ш$;

$y_2 \text{ Ш} \wedge y_{10}^П \rightarrow y_{11}^З \wedge \text{Зпш}(X) \vee y_{11}^ПН \wedge \text{Ппш}(X) = 1$.

Ознака аффрикативності звуку y_{12} :

$\text{Неафпш}(X) = x^Д \vee x^Ж \vee x^ДЖ \vee x^ДЗ \vee x^З \vee x^С \vee x^Т \vee x^Ш$;

$\text{Афпш}(X) = x^Ч \vee x^Ц$;

$y_2 \text{ Ш} \wedge y_{10}^П \rightarrow y_{12}^АФ \wedge \text{Афпш}(X) \vee y_{12}^НЕАФ \wedge \text{Неафпш}(X) = 1$.

Рівняння зв'язку фонем X з ознаками y_{10}, y_{11}, y_{12} :

$z^3 \rightarrow y_{10}^Г x^Б \vee y_{10}^З x^Г \vee y_{10}^П [y_{11}^З (y_{12}^НЕАФ x^Д \vee y_{12}^АФ x^Ц) \vee y_{11}^ПН x^Ч] = 1$;

$z^4 \rightarrow y_{10}^Г x^В \vee y_{10}^З x^Г \vee y_{10}^П (y_{11}^З x^З \vee y_{11}^ПН x^Ж) = 1$;

$z^5 \rightarrow y_{10}^Г x^П \vee y_{10}^З x^K \vee y_{10}^П [y_{11}^З (y_{12}^НЕАФ x^Т \vee y_{12}^АФ x^Ц) \vee y_{11}^ПН x^Ч] = 1$;

$z^6 \rightarrow y_{10}^Г x^Ф \vee y_{10}^З x^Х \vee y_{10}^П (y_{11}^З x^С \vee y_{11}^ПН x^Ш) = 1$.

Створені моделі зв'язують фонему з набором ознак звуку й дозволяють визначити значення фонем в залежності від значень ознак. Уведено відношення, що пов'язує фонетичний запис тексту з його орфографічним записом, між фонемою та фонетичними ознаками звуку. Перехід від фонетичного запису тексту до орфографічного й навпаки здійснено на основі правил української мови, що регулюють вимову (фонетичні правила) та написання (орфографічні правила).

Формально описано відношення, які описують вплив фонетичного контексту на вибір букв для позначення голосних звуків.

Правила фонетичних позицій голосних звуків у слові –

– положення після різних, відносно категорії твердості – м'якості приголосних:

$$G_1^T = J_2(p_V) q_V^T; \quad G_2^M = J_2(p_V) q_V^M,$$

де $J_2(p_V) = p_V^B \vee p_V^V \vee p_V^D \vee p_V^3 \vee p_V^L \vee p_V^M \vee p_V^Г \vee p_V^{ДЖ} \vee p_V^{ДЗ} \vee p_V^H \vee p_V^П \vee p_V^Р \vee p_V^С \vee p_V^Т \vee p_V^Ф$;

– на абсолютному початку слова; після голосних звуків:

$$G_3^H = p_V;$$

$$G_4^Г = p_V^A \vee p_V^Y \vee p_V^O \vee p_V^I \vee p_V^И \vee p_V^E;$$

– після [ц] та шиплячих приголосних:

$$G_5^Ц = p_V^Ц q_V^T;$$

$$G_6^Ш = (p_V^Ш \vee p_V^Ж) (q_V^M \vee q_V^T) \vee p_V^Ч q_V^M;$$

– положення після задньоязикових звуків та після [j]:

$$G_7^3 = p_V^Г \vee p_V^K \vee p_V^X;$$

$$G_8^J = p_V^J.$$

Запис правил рівняннями АСП:

$$\Phi_{11}^T = G_1^T \rightarrow (p^A A \vee p^Y Y \vee p^O O \vee p^I I \vee p^И И) q_V^T t^3;$$

$$\Phi_{12}^T = G_1^T \rightarrow (p^Y Y \vee p^И И) q_V^T (t^1 \vee t^2),$$

де Φ_{11}^T , Φ_{12}^T – буквофонемні відношення твердих наголошених голосних після твердих приголосних; t^1 , t^2 – ознаки ненаголошених та переднаголошених складів відповідно; t_3 – ознака наголошених складів.

$$\Phi_{13}^T = G_1^T p_V \rightarrow p^E q^T t^3 x^E;$$

$$\Phi_{14}^T = G_1^T A \rightarrow p^E q^T (t^1 \vee t^2) x^E;$$

$$\Phi_{15}^T = G_1^T J_2(p_U) \rightarrow p^E q^T t^3 x^E, (x^E \vee x^E),$$

де $A = (p_V^T \vee p_V^D \vee p_V^C \vee p_V^3 \vee p_V^P \vee p_V^H) q_V^T$;

$$\Phi_{21}^M = G_2^M \rightarrow q^M t^3 (p^A Y \vee p^Y Y \vee p^O I \vee p^E E \vee p^И И);$$

$$\Phi_{22}^M = G_2^M \rightarrow p^Y q_V^M x^O (t^2 \vee t^1);$$

$$\Phi_{31}^П = G_3^П \rightarrow (p^A A \vee p^O O) q^T t^3;$$

$$\Phi_{32}^П = G_3^П \rightarrow [(p^Y Y \vee p^E E) q^T \vee p^I q^M x^I] (t^2 \vee t^3);$$

$$\Phi_{41}^Г = G_4^Г \rightarrow [(p^A A \vee p^O O \vee p^Y Y) q^T \vee p^I q^M x^I] t^3;$$

$$\Phi_{42}^Г = G_4^Г \rightarrow p^E t^3 (p^I q^T x^E \vee p^И q^T x^E);$$

$$\Phi_{51}^Ш = G_5^Ш \rightarrow (p^A \sim x^A) (p^E \sim x^E) (p^I \sim x^I) t^3;$$

$$\Phi_{52}^Ш = G_5^Ш \rightarrow (p^Y \sim x^Y) (t^3 \vee t^2 \vee t^1);$$

$$\Phi_{53}^Ш = G_5^Ш p_V \rightarrow p^O x^O q^T t^1;$$

$$\Phi_{54}^Ш = G_5^Ш J^3(p_V) \rightarrow p^O x^O q^T t^3,$$

де предикат $J^3(p_U) = p_U^B \vee p_U^V \vee p_U^Г \vee \dots \vee p_U^Ш$ – формалізує поняття приголосного звуку;

$$\Phi_{61}^Ц = G_6^Ц \rightarrow (p^A A \vee p^O O \vee p^E E) q^T t^3 \vee p^Y q^T x^Y.$$

$$\Phi_{62}^Ц = G_6^Ц \rightarrow [p^I \sim (x^I \sim x^И)] (t^3 \vee t^2 \vee t^1);$$

$$\Phi_{71}^3 = G_7^3 \rightarrow \{ [p^A q^T x^A \vee q^M x^Y] \vee p^O (q^T x^O q^M x^I) t^3 \vee p^Y (q_V^T x^Y \vee q_V^M x^Ю) (t^3 \vee t^2 \vee t^1);$$

$$\Phi_{72}^3 = G_7^3 \rightarrow (p^E x^E \vee p^I x^I) q^M t^3; \quad \Phi_{81}^J = G_8^J \rightarrow (p^A Y \vee p^O I \vee p^E x^E) q^M t^3; \quad \Phi_{82}^J = G_8^J \rightarrow (p^Y \sim x^Ю) q^M.$$

Отримана система рівнянь є підмоделлю буквофонемних зв'язків, що охоплює клас голосних звуків української мови.

Формально описано відношення, які описують вплив фонетичного контексту на вибір букв для позначення приголосних звуків.

Предикати впливу фонетичного контексту: $G_9^{СП}$ – сильна позиція; $G_{10}^{КЦ}$ – кінець позиції; $G_{11}^{ГЛУХ}$ – позиція перед глухим приголосним; $G_{12}^{ДЗ}$ – позиція перед дзвінкими шумовими приголосними.

$$G_9^{СП} = p_U^J \vee J_4(p_U) \vee J_5(p_U) \vee p_U^B,$$

де $J_4(p_U)$ і $J_5(p_U)$ – предикати, що задають формальний опис понять «голосний звук» та «сонорний звук» відповідно й визначаються виразами

$$J_4(p_U) = p_U^A \vee p_U^E \vee p_U^I \vee p_U^O \vee p_U^Y \vee p_U^И;$$

$$J_5(p_U) = p_U^Л \vee p_U^М \vee p_U^Н \vee p_U^Р.$$

$$G_{10}^{КЦ} = p_U; \quad G_{11}^{ГЛУХ} = J_6(p_U); \quad G_{12}^{ДЗ} = J_7(p_U),$$

де $J_6(p_U)$ і $J_7(p_U)$ визначають класи дзвінких та глухих шумових приголосних:

$$J_6(p_U) = p_U^B \vee p_U^V \vee p_U^Г \vee p_U^Д \vee p_U^Ж \vee p_U^3 \vee p_U^{ДЖ} \vee p_U^{ДЗ};$$

$$J_7(p_U) = p_U^П \vee p_U^Х \vee p_U^K \vee p_U^Т \vee p_U^Ш \vee p_U^Ч \vee p_U^С \vee p_U^Ц \vee p_U^Ф.$$

Чотири групи ознак, які формально подамо предикатами:

– предикат $\Phi_1^{СП}$ фіксує залежність між значеннями різних ознак умови, що звук U перебуває у сильній позиції; $\Phi_2^{КЦ}$ – за умови, що Y звучить в кінці слова; $\Phi_3^{ГЛУХ}$ – перед глухими приголосними; $\Phi_4^{ДЗ}$ – перед дзвінкими шумовими приголосними.

Ознака дзвінкості – глухості:

– в сильній позиції регулюються предикатом:

$$\Phi_1^{СП} = G_9^{СП} \rightarrow (p^B x_1^B \vee p^V x_1^V \vee p^Г x_1^Г \vee p^Д x_1^Д \vee p^Ж x_1^Ж \vee p^3 x_1^3 \vee p^K x_1^K \vee p^П x_1^П \vee p^С x_1^С \vee p^Т x_1^Т \vee p^Ф x_1^Ф \vee p^Ш x_1^Ш);$$

– в положенні кінця слова:

$$\Phi_2^{КЦ} = G_{10}^{КЦ} \rightarrow A \vee (p^K \sim (x_1^K \vee x_1^Г)),$$

де $A = (p^П \sim (x_1^П \vee x_1^Б)) \wedge \wedge (p^Ф \sim (x_1^Ф \vee x_1^В)) (p^Т \sim (x_1^Т \vee x_1^Д)) (p^С \sim x_1^3 \vee x_1^С) (p^Ш \sim (x_1^Ш \vee x_1^Ж))$;

– група правил, що задає відношення $\Phi_{3i}^{ГЛУХ}$ (за умови $G_{11}^{ГЛУХ} = 1$):

$$\Phi_{31}^{ГЛУХ} = (p_U^T \vee p_U^C) q_U^M \rightarrow A q^M;$$

$$\Phi_{32}^{ГЛУХ} = G_{11}^{ГЛУХ} p_U \rightarrow (p^П q^T \sim (x_1^П \vee x_1^Б));$$

$$\Phi_{33}^{ГЛУХ} = G_{11}^{ГЛУХ} p_U \rightarrow p^Ф q^T \sim x_1^K \vee x_1^В \vee x_1^Х \vee x_1^Ф;$$

$$\Phi_{34}^{ГЛУХ} = G_{11}^{ГЛУХ} p_U \rightarrow (p^П q^T \sim (x_1^Т \vee x_1^Д));$$

$$\Phi_{35}^{ГЛУХ} = (p_U^T \vee p_U^Ч) q_U^M \rightarrow (p^Т q^M \sim (x_1^Т \vee x_1^Д));$$

$$\Phi_{36}^{ГЛУХ} = G_{11}^{ГЛУХ} p_U \rightarrow (p^K q^T \sim (x_1^K \vee x_1^Г));$$

$$\Phi_{37}^{ГЛУХ} = G_{11}^{ГЛУХ} p_U \rightarrow (p^Ш q^T \sim (x_1^Ш \vee x_1^Ж));$$

$$\begin{aligned} \Phi_{38} \text{ГЛУХ} &= (p_U^K \vee p_U^Ц) q^T \rightarrow (p^C q^T (x_1^C \vee x_1^3)); \\ \Phi_{39} \text{ГЛУХ} &= (p_U^K q_U^T \vee p_U^Ц q_U^M) \rightarrow (p^C q^M (x_1^C \vee x_1^3)); \\ \Phi_{310} \text{ГЛУХ} &= p_U^П q_U^T \rightarrow p^П q^T x_1^П; \\ \Phi_{311} \text{ГЛУХ} &= p_U^Ф q_U^T \rightarrow p^Ф q^T x_1^Ф; \\ \Phi_{312} \text{ГЛУХ} &= p_U^Д q_U^T \rightarrow p^Д q^T x_1^Д; \\ \Phi_{313} \text{ГЛУХ} &= p_U^Т \rightarrow p^Т x_1^Т; \\ \Phi_{314} \text{ГЛУХ} &= (p_U^C \vee p_U^В \vee p_U^Ф \vee p_U^П \vee p_U^М \vee p_U^Т \vee p_U^Х) \\ &\rightarrow p^C q^T x_1^C; \\ \Phi_{315} \text{ГЛУХ} &= (p_U^Г \vee p_U^Ч \vee p_U^Ш) \rightarrow p^C q^M x_1^C; \\ \Phi_{316} \text{ГЛУХ} &= p_U^Ж \rightarrow p^Ж q^T x_1^Ж; \\ \Phi_{317} \text{ГЛУХ} &= p_U^Ш \rightarrow p^Ш q^T x_1^Ш; \\ \Phi_{318} \text{ГЛУХ} &= p_U^Ш \rightarrow p^Ш q^T x_1^Ш; \end{aligned}$$

– склад предиката $\Phi_4^{ДЗ}(X, Y, Z)$ (звук [y] перебуває у фонетичній позиції перед дзвінками шумовими приголосними):

$$\begin{aligned} \Phi_{41}^{ДЗ} &= p^B q^T (p_U^Г \sim x_1^Ф) (p_U^Г \sim x_1^В); \\ \Phi_{42}^{ДЗ} &= (p_U^В \vee p_U^Б) q_U^M \rightarrow p^B q^M x_1^В; \\ \Phi_{43}^{ДЗ} &= (q^T G_{12}^{ДЗ} p_U^Д \vee q^M p_U^Д q_U^М) \rightarrow (p^Д \sim (x_1^Д \vee x_1^Т)); \\ \Phi_{44}^{ДЗ} &= p_U^Б \sim x_1^Д p^Д q^М; \\ \Phi_{45}^{ДЗ} &= (p_U^Б \vee p_U^Д \vee p_U^З \vee p_U^Г) q_U^М \sim p^З q^Т (x_1^З \vee x_1^С); \\ \Phi_{46}^{ДЗ} &= (p_U^Б q_U^М \vee p_U^Д q_U^Т \vee p_U^Г q_U^М) \sim p^З q^М (x_1^З \vee x_1^С); \\ \Phi_{47}^{ДЗ} &= G_{12}^{ДЗ} \rightarrow p^Г q^Т (x_1^Г \vee x_1^Х); \\ \Phi_{48}^{ДЗ} &= p^Б \sim x_1^Б. \end{aligned}$$

Наведена вище система рівнянь є формальним еквівалентом зв'язків, що існують між звуками сонорних, парних і позапарних дзвінких та глухих приголосних і графічними еквівалентами мови, що їм відповідають.

За допомогою запропонованого опису буквофонемних зв'язків у вигляді системи рівнянь алгебри предикатів можна здійснити формальний перехід від фонетичного запису слова до його графічного подання (задача аналізу), за заданим орфографічним текстом отримувати його фонетичну транскрипцію (задача синтезу).

Отримані формальні моделі наглядно показують, що математичний апарат алгебри скінченних предикатів й вибрана методика формального опису дозволяє отримати їх компактний формальний опис, а також опис акцентних характеристик відмінюваних і дієвідмінюваних частин української мови. Такі моделі відповідають двом основним вимогам формалізації, а саме: достатньо точно описують вихідний мовний об'єкт та можуть бути реалізовані засобами обчислювальної техніки. Розроблені формальні моделі акцентних залежностей мови можна використати для економного запису словарних структур природної мови в лексикографічних базах даних.

6. Морфологічний аналіз

У цьому розділі висвітлено шляхи математичного моделювання морфологічного рівня української мови. Об'єктом формального опису є письмова, тобто орфографічна, а не фонематична реалізація словоформ української мови. При цьому окремою словоформою є відтинок тексту між двома проміжками.

Особливу роль під час словозміни відіграють закінчення словоформ (флексії), що передають, як правило, граматичні значення в українських словах. У морфології української мови існує певна залежність (відношення) між флексією та оточуючим її текстом. Завдання полягає у математичному описі існуючої залежності, тобто у формалізації поняття флексії. Текст, що охоплює закінчення, є неоднорідним щодо нього. Розрізняємо найближчий текст, що межує безпосередньо з закінченням у словоформі, і подальший текст, що межує зі словоформою, у якій перебуває закінчення, а також зосередимо подальший розгляд на методиці формального опису процесів зміни слів за допомогою флексій, що можна застосувати й до інших лінгвістичних одиниць тексту.

Дію тексту, що оточує закінчення (як ближчого, так і подальшого), замінимо введенням набору ознак, який проінтерпретуємо як смисл закінчення. Під смислом розумітимемо набір проміжних змінних, що поєднані рівняннями, які описують закінчення, з рівняннями, що описують решту частини тексту. Набір значень цих змінних інтерпретуваємо як значення смислу. Смисл розглядається як змінний вектор $X = (x_1, x_2, \dots, x_m)$, де m – число компонентів вектора смислу, а змінні x_1, x_2, \dots, x_m – компоненти смислу.

Переважна більшість закінчень містить одно-, дво-, трибуквені закінчення та нульову флексію. Це дало підставу для формального подання закінчення у вигляді трикомпонентного вектора $Y = (y_1, y_2, y_3)$, де y_1, y_2, y_3 – літери, що стоять зліва направо на першому, другому та третьому місцях закінчення відповідно.

Для того щоб отримати компактний та повний опис фрагментного морфологічного відношення $L(X, Y)$, необхідно вибрати компоненти вектора X з урахуванням внутрішньої структури мови. Через те, що об'єкт, який моделюється, є достатньо складною для математичного опису системою, зручно розбити його на ряд підсистем, які надалі називатимемо «мовними блоками», що пізніше об'єднуються теоремою про розкладання в алгебрі скінченних предикатів.

Для кожного мовного блоку фіксують значення деяких із компонентів смислу, які ми назвемо обмеженими. Другу групу ознак, що служать безпосередньо для вибору закінчення всередині деякого мовного блока, назвемо змінними. Обмежувальним ознакам поставимо у відповідність змінну ξ ,

а змінним – X . Таким чином, маємо можливість уточнити структуру вектора смислу $X = (\xi, X) = (\xi_1, \xi_2, \dots, \xi_k, x_1, x_2, \dots, x_m)$, де k, m – число обмежувальних та змінних компонентів вектора смислу відповідно.

Змінні ознаки в свою чергу неоднорідні щодо закінчення. Впливові ближнього тексту на закінчення відповідатимуть лексичні ознаки, а подальшого – синтаксичні.

Отже, завдання полягає у формальному описі засобами АСП фрагментарного морфологічного відношення $L = (\xi_1, \xi_2, \dots, \xi_k, x_1, x_2, \dots, x_m, y_1, y_2, \dots, y_n)$, що буде моделлю флексійного оброблення українських слів. Морфологічне відношення визначається своєю характеристичною функцією $L_\varphi(X, Y)$.

Звуження обсягу математичного опису здійснено введенням шести граматичних ознак. Спочатку введено ознаку ξ_1 виду морфеми зі значеннями: закінчення (з), не закінчення (н) (префікс, корінь, суфікс, тощо).

Ознака змінюваності слова, експлікована змінною ξ_2 , може приймати одне з двох значень – змінюване (z_m), незмінюване (n_{3m}).

Введення змінної ξ_3 , що відповідає ознаці типу словозміни, дозволило окремо описувати відмінювання іменних частин мови ($\xi_3 = i$) та відмінювання дієслів ($\xi_3 = d$).

Іменні частини мови розрізняються за ознакою типу відмінювання (змінна ξ_4). Змінна ξ_4 визначена на множині значень: субстантивна (с), ад'єктивна (а), займенникова (з) та числівникова (ч).

Введемо граматичну ознаку змінної іменної частини мови (ξ_5), що може приймати одне з таких значень: іменник (і), прикметник (п), займенник (з), числівник (ч), дієприкметник (дп).

Шоста граматична ознака, що забезпечує звуження об'єкта математичного моделювання, ξ_6 – вид словоформи зі значеннями: регулярна (р), нерегулярна (н).

Проведені лінгвістичні дослідження показали, що необхідно ввести дев'ятнадцять граматичних ознак: x_1 – відмінок зі значеннями: називний (н), родовий (р), давальний (д), знахідний (з), орудний (о), місцевий (м), кличний (кл); x_2 – рід зі значеннями: чоловічий (ч), жіночий (ж), середній (с); x_3 – число зі значеннями: одиниця (о), множина (м); x_4 – ознака одухотвореності зі значеннями: одухотворена (о), не одухотворена (н); x_5 – варіант закінчення зі значеннями: архаїчне (а), сучасне (с); x_6 – ознака наголошеності закінчення зі значеннями: наголошене (н), ненаголошене (н'); x_7 – остання буква основи зі значеннями, заданими на множині букв та знаків українського алфавіту; x_8 – вид основи зі значеннями: тверда (т), м'яка (м); x_9 – вимога встановлення знака «над буквою» \tilde{i} зі значеннями: ставити (с), не ставити (н); x_{10} – повнота словоформи зі значеннями: повна (п), коротка (к); x_{11} – ознака репрезентації зі значен-

нями: інфінітив (і), особова форма (о); x_{12} – спосіб зі значеннями: дійсна (д), наказова (н); x_{13} – часу зі значеннями: минуле (м), не минуле (m^{-3}); x_{14} – особи зі значеннями: перша (1), друга (2), третя (3); x_{15} – тип впливу дієслівної основи зі значеннями: перший (1), другий (2); x_{16} – відмінювання зі значеннями: перше (1), друге (2); x_{17} – ознака наявності частки “ви” зі значеннями: так (т), ні (н); x_{18} – передостання буква основи зі значеннями, заданими на множині більшості букв українського алфавіту; x_{19} – наявність нормативних обмежень зі значеннями: так (т), ні (н).

Лінгвістичні дослідження формальних моделей словозміни української мови виявили специфічний характер індивідуального морфологічного предиката $L_i(X, Y)$, який можна зобразити у вигляді кон'юнкції двох предикатів, що залежать від меншого числа змінних. Лінгвістичною основою такого зображення предиката $L_i(X, Y)$ є різна природа змінних ознак X , що відповідають двом механізмам: синтаксичного впливу тексту та впливу закінчень основ.

Предикати L_{i1} та L_{i2} змістовно інтерпретуються як таблиці впливу основ та парадигматичні таблиці синтаксичного впливу відповідно. Функція $y_j = F(X)$, ($j = \overline{1,3}$), що відповідає предикатові $L_i(X, Y)$, задана на декартовому добутку $A \times B$ значеннями в M у вигляді:

$$Y_j = \varphi(t, s), \text{ де } t = \eta(X_I), s = \mu(X_{II});$$

$$A = X_I \in \{x_1, x_2, \dots, x_5, x_{10}, x_{11}, \dots, x_{14}\};$$

$$B = X_{II} \in \{x_6, x_7, x_8, x_{15}, x_{16}, \dots, x_{19}\}.$$

Для отримання функцій $s = \mu(X_{II})$ та $t = \eta(X_I)$ розроблено відповідні алгоритми. Множина цих номерів прийнята як області значення функції $s' = \mu(X_{II})$:

$$s^1 = \text{ПРГ}(x_7) x_8^t \vee \text{ЗЯЗ}(x_7) \vee \text{ГОЛ}(x_7) \vee x_7^H;$$

$$s^2 = \text{ПРГ}(x_7) x_8^M; \quad s^3 = \text{ГОЛ}(x_7) \vee x_7^H.$$

Отримано два класи суміжності, перенумеровані довільно та прийняті як значення функції $t' = \eta(X_I)$:

$$t^1 = (x_1^H \vee x_1^O \vee x_4^H) x_2^M x_3^O;$$

$$t^2 = (x_1^P \vee x_1^3 \vee x_4^O) x_2^M x_3^O.$$

Предикат $L_{14}(X, Y)$ відповідає функції $y_j = F_i(X)$, що відображає залежність закінчень регулярних слів іменних частин мови ад'єктивного відмінювання від смислу. Подамо цю функцію у вигляді:

$$y_j = \varphi_1(s_1, t_1), \text{ де } s_1 = \mu_1(X_{II}), t_1 = \eta_1(X_I);$$

$$X_I = \{x_1, x_2, x_3, x_4, x_5, x_{10}\}, X_{II} = \{x_6, x_7, x_8\}.$$

Парадигматична таблиця ад'єктивного відмінювання є квадратною матрицею, а таблиця впливу основ – прямокутною. У парадигматичній таблиці класи суміжності результуючого розбиття Q_j' перенумеровані довільним чином.

Способи впливу подальшого тексту на закінчення регулярних словоформ за числом класів суміжності:

$$\begin{aligned}t_1^1 &= (x_1^h \vee x_1^3 x_4^h) x_2^u x_3^o x_{10}^p; \\t_1^2 &= ((x_1^p \vee x_1^3 x_4^o) \wedge x_2^u x_3^o \vee x_1^p x_2^c x_3^o) x_{10}^p; \\t_1^3 &= x_1^d (x_2^u \vee x_2^c) x_3^o x_{10}^p; \\t_1^4 &= (x_1^o (x_2^u \vee x_2^c) x_3^o) x_{10}^p; \\t_1^5 &= x_1^m (x_2^u \vee x_2^c) x_3^o x_{10}^p; \\t_1^6 &= x_1^h x_2^j x_3^o x_{10}^p; \\t_1^7 &= (x_1^p \vee x_1^d \vee x_1^o x_5^a \vee x_1^m) x_2^j x_3^o x_{10}^p; \\t_1^8 &= x_1^3 x_2^j x_3^o x_{10}^p; \\t_1^9 &= x_1^o x_2^j x_3^o x_5^c x_{10}^p; \\t_1^{10} &= (x_1^h \vee x_1^3) x_2^c x_3^o x_{10}^p; \\t_1^{11} &= (x_1^h \vee x_1^3 x_4^h) x_3^m x_{10}^p; \\t_1^{12} &= (x_1^p \vee x_1^a x_4^o \vee x_1^m) x_3^m x_{10}^p; \\t_1^{13} &= x_1^o x_3^m x_{10}^k; \\t_1^{14} &= x_2^u x_3^o x_{10}^k; \\t_1^{15} &= x_2^c x_3^o x_{10}^k; \\t_1^{16} &= x_2^c x_3^o x_{10}^k; \\t_1^{17} &= x_3^m x_{10}^k;\end{aligned}$$

$$\begin{aligned}t_1^1 &\rightarrow (y_1^h \vee y_1^i \vee y_1^o) y_2^j y_3^u; \quad t_1^2 \rightarrow (y_1^a \vee y_1^j) y_2^m y_3^h; \\t_1^3 &\rightarrow (y_1^o \vee y_1^e) y_2^b y_3^i; \quad t_1^4 \rightarrow (y_1^h \vee y_1^i) y_2^m y_3^u; \\t_1^5 &\rightarrow (y_1^o \vee y_1^e) y_2^m y_3^u; \quad t_1^6 \rightarrow (y_1^b \vee y_1^j) y_2^u y_3^m; \\t_1^7 &\rightarrow (y_1^h \vee y_1^i) y_2^j y_3^u; \quad t_1^8 \rightarrow (y_1^o \vee y_1^e) y_2^u y_3^j; \\t_1^9 &\rightarrow (y_1^o \vee y_1^b) y_2^u y_3^j; \quad t_1^{10} \rightarrow (y_1^a \vee y_1^j) y_2^u y_3^x; \\t_1^{11} &\rightarrow (y_1^a \vee y_1^e) y_2^u y_3^p; \quad t_1^{12} \rightarrow (y_1^a \vee y_1^j) y_2^t y_3^u; \\t_1^{13} &\rightarrow (y_1^h \vee y_1^i) y_2^m y_3^h; \\t_1^{14} &\rightarrow (y_1^u (y_2^u (y_3^u \vee y_3^b) \vee y_2^u y_3^j); \\t_1^{15} &\rightarrow (y_1^a \vee y_1^j) y_2^u y_3^u; \\t_1^{16} &\rightarrow (y_1^o \vee y_1^e) y_2^u y_3^u; \quad t_1^{17} \rightarrow (y_1^i \vee y_1^j) y_2^u y_3^u.\end{aligned}$$

Способи впливу ближчого тексту на закінчення регулярних словоформ за числом класів суміжності:

$$\begin{aligned}s_1^1 &= \text{ПРГ}(x_7) x_6^h x_8^t; \quad s_1^2 = \text{ПРГ}(x_7) x_6^h x_8^m; \\s_1^3 &= \text{ЗЯЗ}(x_7) x_6^h; \quad s_1^4 = \text{ШИП}(x_7) x_6^h; \quad s_1^5 = x_6^h x_8^h;\end{aligned}$$

$$s_1^6 = x_6^h (\text{ПРГ}(x_7) \vee x_7^h \vee \text{ГОЛ}(x_7) \vee x_7^h);$$

$$s_1^7 = x_6^h (\text{ЗЯЗ}(x_7) \vee \text{ШИП}(x_7));$$

$$s_1^8 = x_6^h (\text{ГОЛ}(x_7) \vee x_7^h).$$

Функція $y_j = \varphi_1(s_1, t_1)$ відповідності структури закінчення (предикативі L_{14}):

— для першої букви закінчення:

$$y_1^a = (t_1^6 \vee t_1^{15})(s_1^1 \vee s_1^3 \vee s_1^4 \vee s_1^5 \vee s_1^6 \vee s_1^7);$$

$$y_1^j = (t_1^6 \vee t_1^{15})(s_1^2 \vee s_1^8);$$

$$y_1^h = (t_1^4 \vee t_1^{11} \vee t_1^{12} \vee t_1^{13} \vee t_1^{17})(s_1^1 \vee s_1^5 \vee s_1^6) \vee t_1^1 (s_1^1 \vee s_1^5);$$

$$y_1^i = (t_1^4 \vee t_1^{11} \vee t_1^{12} \vee t_1^{13} \vee t_1^{17})(s_1^2 \vee s_1^3 \vee s_1^4 \vee s_1^7 \vee s_1^8) \vee t_1^1 (s_1^2 \vee s_1^3 \vee s_1^4 \vee s_1^8);$$

$$y_1^o = (t_1^2 \vee t_1^3 \vee t_1^5 \vee t_1^7 \vee t_1^9 \vee t_1^{10} \vee t_1^{16})(s_1^1 \vee s_1^3 \vee s_1^6 \vee s_1^7) \vee t_1^1 (s_1^6 \vee s_1^7);$$

$$y_1^e = (t_1^2 \vee t_1^3 \vee t_1^5 \vee t_1^7 \vee t_1^9 \vee t_1^{10} \vee t_1^{16})(s_1^2 \vee s_1^4 \vee s_1^5 \vee s_1^8);$$

$$y_1^y = t_1^8 (s_1^1 \vee s_1^3 \vee s_1^4 \vee s_1^5 \vee s_1^6 \vee s_1^7);$$

$$y_1^j = t_1^8 (s_1^2 \vee s_1^8);$$

$$y_1^u = t_1^{14} (s_1^1 \vee s_1^4 \vee s_1^5);$$

— для другої букви закінчення:

$$y_2^m = t_1^3 \vee t_1^4 \vee t_1^5 \vee t_1^{13};$$

$$y_2^h = t_1^1 \vee t_1^7 \vee t_1^{14} \& s_1^8;$$

$$y_2^x = t_1^{12}; \quad y_2^r = t_1^2;$$

$$y_2^u = t_1^6 t_1^8 \vee t_1^9 \vee t_1^{10} \vee t_1^{11} \vee t_1^{14} (s_1^1 \vee s_1^2 \vee \dots \vee s_1^7) \vee t_1^{15} \vee t_1^{16} \vee t_1^{17};$$

— для третьої букви закінчення:

$$y_3^h = t_1^{13}; \quad y_3^j = t_1^8 \vee t_1^9;$$

$$y_3^a = t_1^6; \quad y_3^e = t_1^{10} \vee t_1^{11};$$

$$y_3^b = s_1^2 \& t_1^{14};$$

$$y_3^o = t_1^2; \quad y_3^y = t_1^3;$$

$$y_3^u = t_1^1 \vee t_1^4 \vee t_1^5 \vee t_1^7 \vee t_1^{12} \vee t_1^{14} (s_1^1 \vee s_1^3 \vee s_1^4 \vee \dots \vee s_1^8) \vee t_1^{15} \vee t_1^{16} \vee t_1^{17}.$$

Формалізація нерегулярних словоформ з нестандартними закінченнями

$$\bigvee_{\langle \beta_1, \beta_2, \dots, \beta_{20} \rangle} \bigwedge_{i=1}^n Z_i^{\beta_i} (y_1 = Z_{n+1}^{\beta_{n+1}}) (y_2 = Z_{n+2}^{\beta_{n+2}}) (y_3 = Z_{n+3}^{\beta_{n+3}}),$$

де $\langle \beta_1, \beta_2, \dots, \beta_{20} \rangle$ — деяке слово з словника системи, $\langle Z_1, Z_2, \dots, Z_{20} \rangle$ — зображення слова зі словника, $\langle y_1, y_2, y_3 \rangle$ — закінчення слова, $\bigvee_{\langle \beta_1, \beta_2, \dots, \beta_{20} \rangle}$ — означає

логічну суму всіх елементів словника, найдовше слово якого містить не більше двадцяти букв.

7. Формалізація принципу уніфікації багатовходових k -значних структур

В узагальненому вигляді універсальні просторові k -значні структури, згідно з узагальненим способом завдання функцій k -значної логіки за допомогою таблиць істинності та принципу симбіозу дво- та багаторівневого кодування і засобів, включають до свого складу такі компоненти: паралельний аналого-цифровий перетворювач (елемент розпізнавання k -значної змінної); дешифратор (ДШ), селектор, комутатор, паралельний цифро-аналоговий перетворювач (ключовий комутатор, або підсумовувач струмів) [5,8].

Висновки

Вирішення задач формалізації принципів організації багатовходових k -значних структур забезпечує побудову новітньої концепції синтезу структур для високошвидкісних обчислювальних систем; застосування просторового та часового паралелізму на структурному й алгоритмічному рівнях та k -значних методів кодування; створення процедурних і функціональних мов, паралельних машин баз знань і логічного виводу [2–5].

Єдино відомим нам об'єктивним носієм знання та інтелекту є людина, а виразником, засобом до зовнішнього спілкування та носієм інтелекту є людська мова, що й складатиме об'єкт і напрямок наступних досліджень.

На підставі вивчення даних фонетики і орфографії української мови розроблено:

— систему фонетичних ознак, а також систему предикатів для формального опису системи фоне-

тичних ознак, що дозволяє здійснювати їх автоматичне оброблення;

– методичку формального опису букво-фонемних відношень, побудовані математичні відношень голосних та приголосних звуків української мови, на основі яких формалізуються закономірності поєднання голосних і приголосних букв та звуків, що утворюють пари за дзвінкістю-глухістю й твердістю-м'якістю, а також поза парних приголосних;

– математичну модель акцентних характеристик української мови. Усі ці результати дозволяють отримати компактну модель для економного запису словників природної мови в пам'яті ЕОМ.

Із допомогою опису букво-фонемних зв'язків у вигляді системи рівнянь АСП можна:

– в інтелектуальному інтерфейсі здійснити формальний перехід від фонетичного запису слова до його графічного зображення (задача фонетичного аналізу);

– за даним орфографічним текстом отримати його фонетичну транскрипцію (задача синтезу мови);

– перейти до системного та структурного синтезу процесорів функціонально-зорієнтованих на аналіз та синтез мови, які є необхідною складовою частиною систем штучного інтелекту.

Наведений перелік характеризує різні рівні роботи з інформацією, виділяє якісно різні групи інформаційних технологій, особливо підкреслюючи можливості роботи із знаннями на рівні контекстно-залежного опису предметної галузі, тобто на рівні семантики і контекстної залежності трактування кожної інформаційної послідовності.

Між першим і подальшими рівнями подання знань проходить стіна, що відокремлює інтелектуальні системи від не інтелектуальних, а саме поняття знань не є винятковою приналежністю інтелектуальних систем і кардинально змінюється на кожному рівні його подання.

Список літератури: 1. *Лачинов, В.М.* Информодинамика или Путь к Миру открытых систем [Текст] / В.М. Лачинов, А.О. Поляков. – Санкт-Петербург: Издательство СПбГТУ. – 1999. – 432 с. 2. *Четвериков, Г.Г.* Формалізація принципів побудови універсальних k -значних структур мовних систем штучного інтелекту [Текст] / Г.Г. Четвериков // Доповіді НАН України. – 2001. – №1 (41). – С. 76–79. 3. *Бондаренко, М.Ф.* Основи теорії синтезу надшвидкодійних структур мовних систем штучного інтелекту [Текст] / М.Ф. Бондаренко, З.Д. Коноплянко, Г.Г. Четвериков. – К.: ІЗМН, 1997. – 264 с. 4. *Бондаренко, М.Ф.* Основи теорії багатозначних структур і кодування в системах штучного інтелекту [Текст] / М.Ф. Бондаренко, З.Д. Коноплянко, Г.Г. Четвериков. – Х.: Фактор-Друк, 2003. – 336 с. 5. *Пат. 2147789 РФ, МКВ Н 03 К 19/02,*

Н 03 М 1/00. Функциональный преобразователь с многозначным кодированием / М.Ф.Бондаренко, З.Д. Коноплянко, Г.Г.Четвериков (Україна). – №97101717/09; Заявл. 04.02.97; Опубл. 24.04.2000, Бюл.№11. – 6 с. 6. *Четвериков, Г.Г.* Алгебро-логічні засоби моделювання природної мови. Повідомлення 1 [Текст] / Г.Г. Четвериков // Бионика интеллекта. – 2004. – №1(61). – С.42–45. 7. *Четвериков, Г.Г.* Алгебро-логічні засоби моделювання природної мови. Повідомлення 2 [Текст] / Г.Г. Четвериков // Бионика интеллекта. – 2005. – №2(62). – С.12–19. 8. *Четвериков, Г.Г.* Алгебро-логічні засоби моделювання природної мови. Повідомлення 3 [Текст] / Г.Г. Четвериков // Бионика интеллекта. – 2007. – №1(66). – С. 48–52. 9. *Широков, В.А.* Очерк основных принципов квантового лингвистики [Текст] / В.А. Широков // Бионика интеллекта. – 2007. – №1(66). – С. 25–32. 10. *Шабанов-Кушнарченко, Ю.П.* Теория интеллекта. Математические средства [Текст] / Ю.П. Шабанов-Кушнарченко. – Харьков: Выща шк., 1984. – 144 с. 11. *Шрейдер, Ю.А.* О понятии “математической модели языка” [Текст] / Ю.А. Шрейдер. – М.: Знание, 1971. – 63 с.

Надійшла до редколегії 12.02.2010 р.

УДК 007. 519.714:519.766

Концепция унификации методов и средств построения параллельных многозначных структур языковых систем / Г.Г. Четвериков // Бионика интеллекта: науч.-техн. журнал. – 2010. – № 1 (72). – С. 3–11.

Статья посвящена анализу проблемы создания систем искусственного интеллекта, которые позволяют моделировать на логическом и аппаратном уровнях различные подходы математического описания естественного языка, и которые являются элементами k -значной структурной организации информационно-интеллектуальных технологий. Показана необходимость и возможность разработки общей теории построения интеллектуального интерфейса и систем искусственного интеллекта, которая могла бы стать методологической основой целеустремленного создания новых интеллектуальных и информационных технологий.

Библиогр.: 11 найм.

UDK 007. 519.714:519.766

The concept of unification methods and construction tools parallel multiple-valued structures language systems / G.G. Chetverikov // Bionics of Intelligence: Sci. Mag. – 2010. – № 1 (72). – P. 3–11.

Article is devoted to analysis problem of creation the artificial intellect systems which allow to model at logic and hardware levels the various approaches of the mathematical description natural language and which are elements of the k -unit structural organization of information-intellectual technologies. Necessity and possibility of working out the general theory of construction the intellectual interface and an artificial intellect systems which could become methodological basis of purposeful creation new intellectual and information technology is demonstrated.

Ref.: 11 items.

УДК 681.3.07



ПОСТРОЕНИЕ 12-ФАКТОРНОЙ МОДЕЛИ ЧЕЛОВЕЧЕСКОГО ЛИЦА

Е.П. Путятин¹, П.А. Орбинский²¹ ХНУРЭ, г. Харьков, Украина, informatika@kture.kharkov.ua² ХНУРЭ, г. Харьков, Украина, orbp@mail.ru

В работе рассмотрены теоретические аспекты распознавания человеческого лица по фотографии, а также результаты разработки алгоритмического и программного обеспечения для автоматического выделения контура и определения основных групп частей лица. Результатом применения предложенных методов является построение 12-факторной модели, описывающей человеческое лицо, позволяющей быстро с заданной точностью идентифицировать личность.

СИСТЕМА ИДЕНТИФИКАЦИИ ЛИЧНОСТИ ЧЕЛОВЕКА, ВЫДЕЛЕНИЕ КОНТУРОВ ЛИЦА, СЕГМЕНТАЦИЯ В ПРОСТРАНСТВЕ HSI, МАТЕМАТИЧЕСКАЯ МОДЕЛЬ, ПОЛУТОНОВОЕ ИЗОБРАЖЕНИЕ, МОРФОЛОГИЧЕСКАЯ ФИЛЬТРАЦИЯ ИЗОБРАЖЕНИЯ

Введение

На сегодняшний день использование видеoinформации в современном мире стремительно возрастает. Наиболее остро стоит проблема распознавания объектов. Одной из важных проблем распознавания является идентификация человека по его фото или видеосъемке.

Целью работы является разработка и анализ алгоритмов автоматического выделения контура и определение основных групп частей лица человека по цветному изображению для построения 12-факторной модели, описывающей человеческое лицо.

Достижение поставленной цели потребовало решения следующих основных задач:

- разработка и анализ алгоритмов автоматического выделения контура и определение основных групп частей лица на цветных изображениях в режиме реального времени;
- построение 12-факторной модели, описывающей человеческое лицо.

Одним из возможных вариантов решения задачи распознавания является распознавание при помощи скрытых марковских моделей (СММ) [1]. В рассмотренной литературе авторы отмечают средний процент распознавания в 85% и максимальный в 96,5% [1].

В работе [2] авторы провели обзор и анализ методов, построенных на применении различных типов нейронных сетей. В своей работе авторы указывают, что в некоторых случаях процент распознавания доходил до 95.

Еще одним применяемым методом является распознавание лиц на основе псевдометрики [3]. Задача распознавания лиц заключается в вычислении меры близости между лицом-запросом и всеми лицами из базы изображений и поиске наиболее близких лиц, либо таких лиц, для которых мера близости не превышает заданного порогового значения. Из достоинств метода следует отметить удовлетворительную надежность распознавания,

при этом что метод не требует нормализации базы лиц и предварительной обработки изображения, по вычислительным затратам сравнимой с самой процедурой распознавания. Главным недостатком является более низкий по сравнению с другими методами процент распознавания.

1. Особенности системы HSI

Глядя на окрашенный объект, человек описывает его с помощью цвета (цветового тона), насыщенности и светлоты. Цветовой тон является характеристикой, которая описывает собственно цвет (чистый желтый, оранжевый, красный и так далее), тогда как насыщенность даст меру того, в какой степени некоторый чистый цвет разбавлен степенью серого. Светлота субъективно соответствует понятию интенсивности (полутонной яркости) в ахроматическом случае и является одним из ключевых параметров для описания цветового восприятия. Как известно, интенсивность (яркость) — основная характеристика монохромных (полутонных) изображений. Эта величина может быть измерена и легко поддается интерпретации. В модели, которая носит название *цветовая модель HSI*, яркостная составляющая I отделена от цветовой составляющей (цветовой тон H , насыщенность S) [4]. В результате модель *HSI* представляет собой идеальное средство для построения алгоритмов обработки изображений, поскольку в основе модели лежит естественное и интуитивно близкое человеку описание цвета, а ведь именно человек, в конечном счете, является и разработчиком, и пользователем этих алгоритмов. Модель RGB (значения яркости красного, зеленого и синего цветов соответственно) хорошо подходит для создания цветных изображений как при их регистрации цветовой камерой, так и при воспроизведении на экране монитора, но весьма ограничена в том, что касается *описания* цвета.

Цветное RGB изображение может рассматриваться как совокупность трех полутонных изобра-

жений, яркость которых соответствует интенсивности красного, зеленого и синего цветов.

В системе HSI тон H задается формулой (1):

$$H = \begin{cases} \theta, B \leq G \\ 360 - \theta, B > G \end{cases} \quad (1)$$

где

$$\theta = \arccos \left(\frac{\frac{1}{2}((R-G) + (R-B))}{\sqrt{(R-G)^2 + (R-B)(G-B)}} \right). \quad (2)$$

Насыщенность S :

$$S = 1 - \frac{3}{R+G+B} \min[R, G, B]. \quad (3)$$

Интенсивность I :

$$I = \frac{1}{3}(R+G+B). \quad (4)$$

2. Алгоритм выделения контура и определения основных групп частей лица

Предполагается, что на изображении присутствовал основной объект (лицо человека) и фон (белая стена), создавая тем самым контраст между лицом и фоном. Кроме того, центр тяжести человеческого лица близок к центру тяжести фотографии.

Для последующей подготовки изображения с целью построения математической модели в основу положена следующая последовательность операций:

- выделение внутреннего контура;
- выделение контуров мелких объектов на захваченной области с помощью детектора краев Кэнни;
- морфологическая фильтрация изображения и аппроксимация полученных контуров;
- обработка и анализ полученных объектов;
- построение 12-факторной модели для метода идентификации человеческого лица.

Для выделения внутреннего контура авторами использовался компонент интенсивности. В качестве порогового значения, по которому производилось выделение контура лица, было решено использовать математическое ожидание – среднеарифметическое значений пикселей интенсивности. Предварительно компонент интенсивности был преобразован с помощью формулы:

$$I' = \{x'_{ij} | x'_{ij} + \max_{\substack{k=i-1, i+1 \\ n=j-1, j+1}} (x_{kn}), x_{kn} \in I\}, \quad (5)$$

где i, j – координаты пикселей изображения, I' – выходное изображение, x'_{ij} – пиксели выходного изображения, x_{ij} – пиксели интенсивности исходного изображения, \max – максимум на заданном множестве – окрестности пикселя 3×3 ; k, n – координаты окрестности пикселя 3×3 .

Априори известно, что значения интенсивности фона меньше значений интенсивности человечес-

кого лица. В результате применения формулы (5) каждый пиксель интенсивности исходного изображения увеличивается на максимальное значение из окрестности 3×3 . Следствием этого является увеличение порогового значения (математического ожидания), а значит при последующем выделении контура порог «преодолеют» только пиксели с максимальным значением, принадлежащие человеческому лицу. Условием контура является равенство нулю второй производной интенсивности по направлению \vec{n} [5]

$$\frac{\partial^2}{\partial^2 \vec{n}} I, \quad (6)$$

где I – сглаженное изображение, а вектор \vec{n} должен быть ориентирован в направлении, перпендикулярном контуру. Поскольку это направление неизвестно априори, оно аппроксимируется направлением градиента:

$$\vec{n} = \frac{\nabla(I)}{|\nabla(I)|}, \quad (7)$$

где

$$\nabla = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y} \right). \quad (8)$$

В качестве точек контура будем выбирать те точки, в которых значение интенсивности контура I (модуль градиента) больше некоторого порога, задаваемого вручную. Аппроксимацию производных произведем с помощью метода Собеля [6], то есть производные считаются как свертки изображения со следующими масками размера 3×3 . Эти маски представлены в табл. 1.

Таблица 1

Вертикальная и горизонтальная маска для метода Собеля

Вертикальный угол			Горизонтальный угол		
-1	0	1	1	-2	-1
-2	0	2	0	0	0
-1	0	1	1	2	1

Для устранения негладкости контура и объединения близких параллельных граней, соответствующих одному исходному контуру (фильтр Кэнни выделяет линейный контур как две параллельные грани), полученные контуры расширяются с помощью бинарной морфологической операции расширения. Кроме этого, в алгоритм включено нахождение линии шеи и ее отсечение.

3. Фильтрация изображения и аппроксимация полученных контуров

Используя алгоритм поиска связанных компонентов на бинарном изображении, получаем несколько множеств – множество пикселей, принадлежащее левому глазу, множество пикселей, принадлежащее правому глазу, множество пикселей, прина-

длежащее носу, рту, а также несколько множеств, принадлежащих помехам. Помехами считаются, например, клоч волос, упавший на лоб, усы или борода, брови. Поскольку при анализе лица, опираясь на исследования ученых [6], 70% отводится форме головы, 20% – глазам и носу, а всему остальному лишь 10%. Так как человек может менять усы и бороду, то не носить их, надевать солнцезащитные очки, которые будут скрывать брови, этими объектами решено было пренебречь. Кроме того, использовался морфологический анализ того или иного объекта – по удаленности от центра тяжести лица, по размеру относительно соседствующих объектов, а также по расположению на лице. Лицо (U) было разделено на 3 части:

- северо-западная часть (U_{nw}) (верхняя левая четверть лица), основной объект – левый глаз оставлялся для дальнейшей обработки по признаку наибольшей площади;
- северо-восточная часть (U_{ne}) (верхняя правая четверть лица), основной объект – правый глаз оставлялся для дальнейшей обработки по тому же признаку;
- южная часть (U_s) (нижняя половина лица), основной объект рот – объект, имеющий наибольшую площадь и максимально приближенный к нижней части лица.

При этом $U = U_{nw} \cup U_{ne} \cup U_s$.

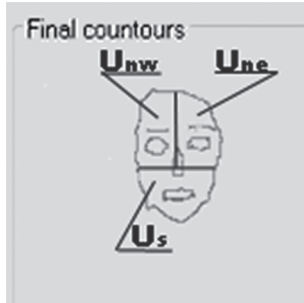


Рис. 1. Разбиение лица на части – северо-западная (U_{nw}), северо-восточная (U_{ne}) и южная (U_s)

Нос выделялся отдельно как фигура, имеющая максимальную площадь, наиболее приближенная к центру тяжести лица, но такая, что ее точки принадлежат множеству всего лица U и не принадлежат ни левому, ни правому, ни рту.

Эти множества можно описать следующим образом:

Лицо – само множество U .

Левый глаз – A :

$$A = \{u_{ij} \in U_{nw} \forall i, j\},$$

$$S_A = \max_k(S_{A_k}), A_k = \{u_{ij} \in U_{nw} \forall i, j\}, \quad (9)$$

где U_{nw} – северо-западная часть лица; A_k – объект, расположенный в северо-западной части лица;

S_{A_k} – площадь объекта A_k ; u_{ij} – пиксели входного изображения; i, j – координаты пикселя.

Правый глаз – B :

$$B = \{u_{ij} \in U_{ne} \forall i, j\}$$

$$S_B = \max_k(S_{B_k}), B_k = \{u_{ij} \in U_{ne} \forall i, j\}, \quad (10)$$

где U_{ne} – северо-восточная часть лица; B_k – объект, расположенный в северо-восточной части лица; S_{B_k} – площадь объекта B_k , u_{ij} – пиксели входного изображения; i, j – координаты пикселя.

Рот – C :

$$C = \{u_{ij} \in U_{ne} \forall i, j\},$$

$$S_C = \max_k(S_{C_k}), C_k = \{u_{ij} \in U_{ne} \forall i, j\}, \quad (11)$$

где U_s – южная часть лица; C_k – объект, расположенный в южной части лица; S_{C_k} – площадь объекта C_k , u_{ij} – пиксели входного изображения; i, j – координаты пикселя.

Нос – D :

$$D = \{u_{ij} \mid u_{ij} \notin C \cap u_{ij} \notin B \cap u_{ij} \notin A\},$$

$$S_D = \max_k(S_{D_k}) \& R[(x_0, y_0), (x_{D_k0}, y_{D_k0})] \rightarrow \min, \quad (12)$$

$$D_k = \{u_{ij} \mid u_{ij} \notin C \cap u_{ij} \notin B \cap u_{ij} \notin A\},$$

где

$$R = \sqrt{(x_{D_k0} - x_0)^2 + (y_{D_k0} - y_0)^2} \quad (13)$$

расстояние от центра тяжести лица (x_0, y_0) до центра тяжести k -го объекта (x_{D_k0}, y_{D_k0}); u_{ij} – пиксели входного изображения; i, j – координаты пикселя.

Лоб – G :

$$G = \{u_{ij} \in U_{ne} \cup U_{nw}\}, \quad (14)$$

где u_{ij} – пиксели входного изображения; i, j – координаты пикселя; G – множество пикселей, принадлежащих северо-восточной или северо-западной части лица, таких, что y -координата любого из них больше y -координаты любого пикселя, принадлежащего левому глазу (множество A) или правому глазу (множество B).

4. Построение 12-факторной модели человеческого лица

Далее все полученные объекты, кроме носа, аппроксимировались эллипсами произвольной формы. Ниже приведен алгоритм аппроксимации.

У произвольного эллипса существует пять неизвестных параметров: (x_0, y_0) – координаты центра, α – угол, определяющий ориентацию, (a, b) – малая и большая полуоси. В нашем случае *априори* известны: ориентация $\alpha = 90^\circ$ (если голова человека при съемке расположена вертикально). Примерно известны центры тяжести каждого из объектов.

Таким образом, нам необходимо определить четыре параметра: (a, b) – малая и большая полуоси, (dx, dy) – смещение центра эллипса по оси x и y со-

ответственно. Для нахождения параметров решается следующая задача максимизации:

$$E = \sum_{u \in U} f(u) \rightarrow \max(S_E), \quad (15)$$

где E – рассматриваемый эллипс; S_E – площадь эллипса, а $f(u)$ задается формулой:

$$f(u) = \begin{cases} 1, u \in E \\ 0, u \notin E \end{cases}. \quad (16)$$

Так как наши параметры являются целыми числами, то задачу максимизации можно достаточно эффективно решать простым перебором [7].

Область, описывающая нос человека, остается без изменений, то есть не аппроксимируется.

Применив формулу (15) к формулам (9) – (11), а также к множеству U , описывающему все лицо, получаем 4 эллипса: $E1$ – эллипс, представляющий левый глаз, $E2$ – эллипс, представляющий правый глаз, $E3$ – эллипс, представляющий лоб, $E4$ – эллипс, представляющий окружность лица, N – многоугольник, представляющий нос, получен из формулы (12).

Построим 12-факторную модель $M(k_i) = \begin{pmatrix} k_1 \\ k_2 \\ k_3 \\ k_4 \\ k_5 \\ k_6 \\ k_7 \\ k_8 \\ k_9 \\ k_{10} \\ k_{11} \\ k_{12} \end{pmatrix}$,

описывающую полученное лицо, предварительно определив расстояние $R(X, Y)$ между двумя точками $X(x_1, y_1)$ и $Y(x_2, y_2)$ как евклидово расстояние. Евклидово расстояние является геометрическим расстоянием в многомерном пространстве (в нашем случае 2-мерное пространство) и имеет определенные преимущества (например, расстояние между двумя объектами не изменяется при введении в анализ нового объекта, который может оказаться выбросом):

$$R(X, Y) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (17)$$

1) $k_1 = \frac{l_{E4}}{d_{E4}}$ – отношение высоты эллипса $E4$ к его ширине, где l_{E4} – высота эллипса $E4$; d_{E4} – ширина эллипса $E4$.

2) $k_2 = \frac{l_{E3}}{d_{E3}}$ – отношение высоты эллипса $E3$ к его ширине, где l_{E3} – высота эллипса $E3$; d_{E3} – ширина эллипса $E3$.

3) $k_3 = \frac{l_{E2}}{d_{E2}}$ – отношение высоты эллипса $E2$ к его ширине, где l_{E2} – высота эллипса $E2$; d_{E2} – ширина эллипса $E2$.

4) $k_4 = \frac{l_{E1}}{d_{E1}}$ – отношение высоты эллипса $E1$ к его ширине, где l_{E1} – высота эллипса $E1$; d_{E1} – ширина эллипса $E2$.

5) $k_5 = \frac{l_{E4}}{l_{E3}}$ – отношение высота эллипса $E4$ к высоте эллипса $E3$, где l_{E3} – высота эллипса $E3$; l_{E4} – высота эллипса $E4$.

6) $k_6 = \frac{d_{E4}}{d_{E3}}$ – отношение ширины эллипса $E4$ к ширине эллипса $E3$, где d_{E3} – ширина эллипса $E3$; d_{E4} – ширина эллипса $E4$.

7) $k_7 = \frac{R(O_{E1}, O_{E3})}{R(O_{E1}, O_{E2})}$ – отношение расстояния между центрами эллипсов $E1$ и $E3$ к расстоянию между центрами эллипсов $E1$ и $E2$, где R определяется по формуле (17), O_{E1}, O_{E2}, O_{E3} – центры эллипсов $E1, E2, E3$ соответственно.

8) $k_8 = \frac{R(O_{E2}, O_{E3})}{R(O_{E1}, O_{E2})}$ – отношение расстояния между центрами эллипсов $E2$ и $E3$ к расстоянию между центрами эллипсов $E1$ и $E2$, где R определяется по формуле (17), O_{E1}, O_{E2}, O_{E3} – центры эллипсов $E1, E2, E3$ соответственно.

9) $k_9 = \frac{R(O_N, O_{E3})}{R(O_{E1}, O_{E2})}$ – отношение расстояния между центрами эллипса $E3$ и многоугольника N к расстоянию между центрами эллипсов $E1$ и $E2$, где R определяется по формуле (17), $O_{E1}, O_{E2}, O_{E3}, N$ – центры эллипсов $E1, E2, E3$ и многоугольника N соответственно.

10) $k_{10} = \frac{R(O_{E1}, O_{E4})}{R(O_{E1}, O_{E2})}$ – отношение расстояния между центрами эллипсов $E4$ и $E1$ к расстоянию между центрами эллипсов $E1$ и $E2$, где R определяется по формуле (17), O_{E1}, O_{E2}, O_{E4} – центры эллипсов $E1, E2, E4$ соответственно.

11) $k_{11} = \frac{R(O_{E2}, O_{E4})}{R(O_{E1}, O_{E2})}$ – отношение расстояния между центрами эллипсов $E4$ и $E2$ к расстоянию между центрами эллипсов $E1$ и $E2$, где R определяется по формуле (17), O_{E1}, O_{E2}, O_{E4} – центры эллипсов $E1, E2, E4$ соответственно.

12) $k_{12} = \frac{R(O_N, O_{E4})}{R(O_N, O_{E3})}$ – отношение расстояния между центрами эллипса $E4$ и многоугольника N к расстоянию между центрами эллипса $E3$ и многоугольника N , где R определяется по формуле (17), O_{E3}, O_{E4}, N – центры эллипсов $E3, E4$ и многоугольника N соответственно.

5. Эксперимент

Исходя из формул (5) – (17), был реализован соответствующий алгоритм и написана программа, результат работы которой приведен на рис. 2–5.

Слева вверху – входное изображение, справа вверху – *H*-компонент, слева внизу – *S*-компонент, справа внизу – *I*-компонент.

Вверху слева – входное изображение, вверху справа – полутоновое изображение компоненты интенсивности, внизу слева – изображение после применения формулы (12), внизу справа – изображение после наложения маски 8-связности.

На рис. 3 приведено изображение, полученное после обработки с использованием формулы (5), медианного фильтра и бинаризации методом Оцу

[8] и маски $\begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$, данная маска позволяет вы-

делять лишь контур изображения.

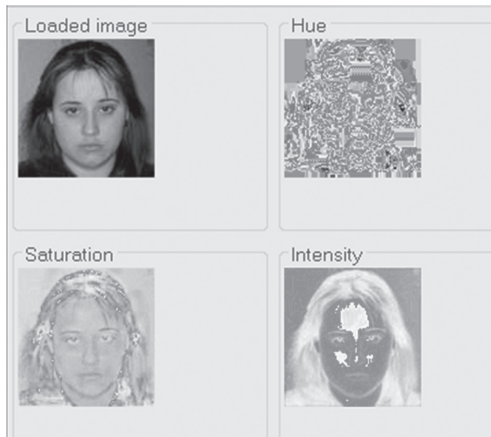


Рис. 2. Пример представления входного изображения в *HSI* системе

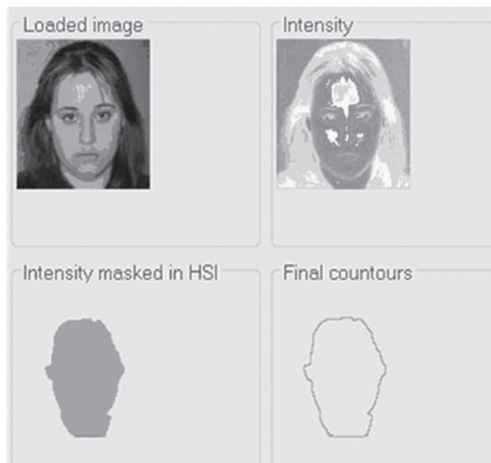


Рис. 3. Изображение после обработки по формуле (15) и фильтрации

Однако, как видно из рис. 2 и 3, для выделения мелких частей лица (глаз, рта, носа) значения интенсивности подходят значительно меньше, нежели значения насыщенности, поэтому для обра-

ботки захваченной области с помощью детектора краев Кэнни наложим полученный контур, как фильтр на значение насыщенности и таким образом получим «чистое» изображение лица.

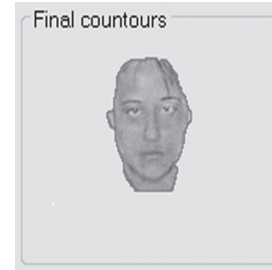


Рис. 4. Изображение после наложения контура лица на компоненту насыщенности

Результат применения выделения краев Кэнни [8] приведен на рис. 5

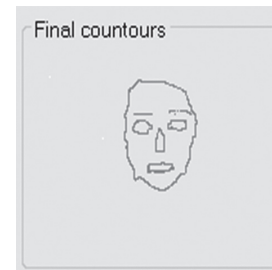


Рис. 5. Результаты применения оператора Кэнни

Повторим эксперимент с тем лишь отличием, что входное изображение не преобразовывается в систему *HSI*. Результат приведен на рис. 6

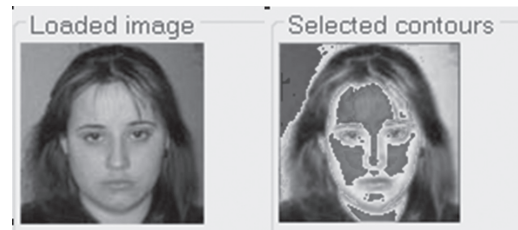


Рис. 6. Изображение после обработки по формуле (5) и фильтрации (в системе RGB)

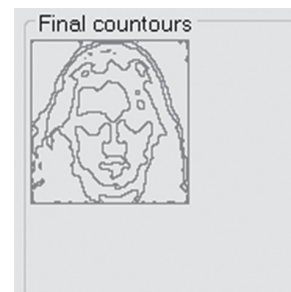


Рис. 7. Результаты применения оператора Кэнни(в системе RGB)

6. Анализ метода

После ряда испытаний на 100 различных фотографиях человеческих лиц, на 90 снимках контур лица был выделен «чисто», на остальных 10 были некоторые неточности. В частности, были захваче-

ны лишние участки фона, не относящиеся к человеческому лицу. Связано это с тем, что эти снимки были более низкого качества, нежели остальные 90. Приведем таблицу зависимости погрешности выделения контура от качества снимка. Все снимки имеют размер 200×200 пикселей и сохранены в формате jpeg.

Таблица 2

Зависимость погрешности выделения контура от качества снимка

Размер файла, (кб)	Качество	Площадь лица, рассчитанная после выделения контура вручную (пиксели)	Площадь лица, рассчитанная после выделения контура программой (пиксели)	Погрешность, %
43	Макс.	12575	12642	0,5328
42	Макс.	11758	11725	0,2806
42	Макс.	10568	10569	0,0094
39	Высокое	10035	10156	1,2057
38	Высокое	11245	11645	3,5571
40	Высокое	10689	10978	2,7037
32	Среднее	11244	11901	5,84311
33	Среднее	11023	12098	6,9037
32	Среднее	12047	11159	4,8310
20	Низкое	11156	12896	15,5969
21	Низкое	11867	13125	10,6008
19	Низкое	10992	9689	11,8541

Из табл. 2 очевидно, что наибольшая погрешность наблюдается на снимках худшего качества.

На основании проведенных экспериментов (рис. 6 и рис. 7) можно говорить о нецелесообразности применять описанный метод для изображения в системе RGB, так как необходимые для последующей идентификации человека объекты (глаза, нос, рот, лоб) выделить не удалось.

Заключение

Предложены и программно реализованы новые методы автоматического выделения контура и определения основных групп частей лица человека на цветных изображениях, устойчивые к зашумлению и яркостному искажению анализируемого изображения. Проанализированы основные подходы к решению задач автоматического выделения контура и определения основных групп частей лица человека. На основе проведенных статистических экспериментов получены оценки надежности рассмотренных алгоритмов автоматического выделения контура и определения основных групп частей лица человека. Построена 12-факторная модель, описывающая человеческое лицо.

Алгоритм был реализован на языке программирования C# и протестирован на большом числе снимков различного качества и освещенности. Программа позволяет с достаточно высокой точностью выделять как четкие, так и размытые контуры на высоко и низкоконтрастных изображениях. Результаты апробации системы показали 90% правильного выделения границ.

Как видно из приведенного описания, выделение контура при помощи указанного метода можно еще существенно улучшить в первую очередь за счет добавления процедуры обучения (цветовая сегментация). Но в то же время простота использования, легкость и скорость распознавания (фотография 200×200 обрабатывается ~ 1 сек на P4-1700), и отсутствие необходимости обучения являются очень привлекательными.

В качестве недостатка данного метода следует отметить необходимость высокого качества входного изображения для точной работы, что, безусловно, осложняет его применение, например в системах видеослежения, где качество съемки невысоко.

Следует отметить и дополнительные временные затраты, необходимые для преобразования входного изображения в системе RGB в систему HSI.

Список литературы: 1. Шапиро, Л. Компьютерное зрение; пер. с англ. [Текст] / Л. Шапиро, Д. Стокман. – М.: БИНОМ. Лаборатория знаний, 2006. – 752 с. 2. Файн, В.С. Опознавание изображений [Текст] / В.С. Файн – М.: Наука, 1970. – 299 с. 3. Гонсалес, Р. Цифровая обработка изображений [Текст] / Р. Гонсалес, Р. Вудс. – М.: Техносфера, 2005 – 1072 с. 4. Wu J. Robust Facial Landmark Detection for Intelligent Vehicle System / J. Wu, M. Trivedi. – San Diego, USA, 2005. 5. Alboil A. An Unsupervised Color Image Segmentation Algorithm for Face Detection Applications. / A. Alboil, L. Torres, E. Delp. – Spain, 2000. 6. Zuo F. Fast Human Face Detection Using Successive Face detectors with Incremental Detection Capability. / F. Zuo. – Eindhoven, the Netherlands, 2002. 7. Гончаров, А. Распознавание лиц на изображениях с низким разрешением [Текст] / А. Гончаров, А. Горбань; – ТТИУФУ, 2007 – 130 с. 8. Информатика и проблема телекоммуникаций // Материалы российской научно-технической конференции. Новосибирск: СибГУТИ, 2005. – Том I. – С. 130-132. 9. Старовойтов, В.В. Нейросетевые методы распознавания изображений [Текст] / В. В. Старовойтов, Брилюк Д. В. http://rusnauka.narod.ru/lib/author/briluk_d_b/1/. 10. Форсайт, Д. А. Компьютерное зрение. Современный подход [Текст] / Д. А. Форсайт, Д. Понс – К.: Вильямс, 2004. – 928 с. 11. Путятин, Е.П. Обработка изображений в робототехнике [Текст] / Е.П. Путятин, С.И. Аверин – М.: Машиностроение, 1990. – 320 с.

Поступила в редколлегию 16.02.2010 г.

УДК 681.3.07

Побудова 12-факторної моделі обличчя людини / Є.П. Путятін, П.А. Оробінський // Біоніка інтелекту: наук.-техн. журнал. – 2010. – № 1 (72). – С. 12-17.

У роботі розглянуто та програмно реалізовано нові методи автоматичного виділення контура та визначення основних груп частин обличчя людини на кольорових зображеннях. Розглянуто теоретичні аспекти розпізнавання обличчя людини за фотографією. Побудовано 12-факторну модель обличчя людини.

Табл. 2. Іл. 7. Бібліогр.: 11 найм.

UDK 681.3.07

Construction of 12-factorial model of a human face / E.P. Putyatin, P.A. Orobinskiy // Bionics of Intelligence: Sci. Mag. – 2010. – № 1 (72). – P. 12-17.

In the given work proposed and discussed new methods for automatic border detection and identify the main parts of human face. The theoretical aspects of human's face recognition by photo were considered. 12-factors model was created.

Tab. 2. Fig. 7. Ref.: 11 items.

УДК 519.62



ОПРЕДЕЛЕНИЕ СТРАТЕГИЙ В ТРЕЙДИНГОВЫХ СИСТЕМАХ НА ОСНОВЕ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ

А.А. Гришко, С.Г. Удовенко, Л.Э. Чалая

ХНУРЭ, г. Харьков, Украина, udovenko@kture.kharkov.ua

В работе проведен анализ методов машинного обучения, применимых к компьютерным системам биржевой торговли. Предложен и протестирован модифицированный метод формирования торговых стратегий, основанный на использовании Q -обучения. Новый подход позволяет пользователю трейдинговой системы непрерывно получать, анализировать и использовать для принятия решений рыночную информацию.

ТРЕЙДИНГОВАЯ СИСТЕМА, Q -ОБУЧЕНИЕ, ТЕХНИЧЕСКИЙ АНАЛИЗ, ИНДИКАТОР, ТОРГОВАЯ СТРАТЕГИЯ

Введение

Развитие информационных технологий на основе применения методов вычислительного интеллекта способствовало появлению компьютерных систем биржевой торговли (трейдинговых систем). Возможности, которыми раньше располагали только крупные инвестиционные компании и банки, стали доступны широкому кругу пользователей сети Интернет, связанных с проблемами сбережения, накопления и инвестирования свободных денежных средств. К наиболее перспективным путям решения этих проблем можно отнести торговлю на бирже через Интернет.

Последние исследования показывают эффективность применения в электронной биржевой торговле методов машинного обучения [1, 2]. Рассмотрим основные принципы, используемые при машинном обучении: обучение по результатам контроля, неконтролируемое обучение и обучение с использованием сигналов подкрепления. В первом случае обучение основано на анализе входного вектора и связанного с ним выходного вектора. Контроль основан на том, что в течение обучения требуемые выходные векторы задаются извне и используются как эталонные данные. Такой контроль наиболее часто применяется для распознавания образов, аппроксимации функций и прогнозирования. При неконтролируемом обучении единственной информацией, доступной обучающей системе является набор наблюдаемых входных образцов. В этом случае система обучается без вмешательства учителя, то есть она основана исключительно на использовании обучающих выборок. Зачастую, несмотря на большой объем обучающих данных, здесь не удается достичь желаемых результатов из-за трудоемкости обучения или сложности поставленной цели.

Принцип машинного обучения, который является наиболее перспективным для создания модели финансового рынка, — обучение с подкреплением (reinforcement learning (RL)). Суть такого обучения сводится к следующему: агент трейдинговой

системы (например, торговый робот) должен исследовать текущие биржевые ситуации и принимать решения даже при неполном знании об этих ситуациях. Единственная обратная связь, получаемая агентом от биржевого рынка — скалярный сигнал подкрепления, который является положительным, если его действия выгодны трейдеру, и отрицательным в противном случае. Задача агента — выбрать свои действия, чтобы увеличить сумму сигналов подкреплений на длительном интервале времени [3]. Кроме сигналов подкрепления агент также получает информацию относительно текущего состояния биржевого рынка (в форме вектора наблюдений).

Для решения поставленных в настоящей статье задач модифицируем RL-метод, основанный на алгоритме Q -обучения, предложенном для частично наблюдаемых марковских процессов в работе [3].

Суть модификации состоит в расширении возможностей исходного алгоритма для его работы в он-лайн режиме на основе данных, получаемых из финансового рынка.

1. Постановка задачи

Целью настоящей работы является разработка и тестирование методов определения стратегий трейдера на электронной бирже на основе использования Q -обучения. В связи с этим были поставлены следующие задачи:

— разработать метод Q -обучения с подкреплением, применимый к электронным финансовым рынкам;

— протестировать процедуру формирования торговых стратегий, основанных на предложенном методе.

Для тестирования оценки эффективности разрабатываемых алгоритмических и программных средств используем данные международного межбанковского валютного рынка FX (Foreign Exchange Market), доступные индивидуальным пользователям.

2. Модифицированный метод Q -обучения

Задача Q -обучения с подкреплением может быть описана марковским процессом решений, который идентифицирует дискретный набор состояний окружающей среды S и выполняет одно из возможных действий из множества A . В ответ на действие a_t в момент t при текущем состоянии среды s_t агент системы получает ответный сигнал подкрепления $r_t = r(s_t, a_t)$ от окружающей среды, после чего окружающая среда переходит в новое состояние $s_{t+1} = \delta(s_t, a_t)$. В алгоритме используются функции перехода $\delta(s_t, a_t)$. Функции перехода и подкрепления зависят только от текущего состояния и действий и не зависят от предыдущих состояний и действий.

Задача базового алгоритма Q -обучения – определить и реализовать стратегию $\pi: S \rightarrow A$, основанную на текущем состоянии s_t ; это может быть записано, как $\pi(s_t) = a_t$. Обычно требуется найти стратегию, соответствующую максимальному значению длительной суммы сигналов подкрепления. Чтобы формализовать это, введем функцию $V^\pi(s_t)$, которая является суммой всех сигналов, полученных алгоритмом, стартующим из состояния s_t с последующей стратегией π :

$$V^\pi \leftarrow \sum_{i=0}^{\infty} \gamma^i \cdot r_{t+i}, \quad (1)$$

где r_{t+i} – последовательность сигналов подкрепления; γ ($0 \leq \gamma \leq 1$) – коэффициент дисконтирования, который определяет текущую оценку будущих доходов. При $\gamma = 0$ алгоритм отдает предпочтение максимизации текущего дохода. При приближении коэффициента дисконтирования к 1, больший вес алгоритм присваивает будущим доходам.

Оптимальная стратегия, максимизирующая полный доход, начиная с любого состояния, может быть представлена в виде:

$$\pi^* \leftarrow \arg \max_{\pi} V^\pi(s), \quad (2)$$

где $s \in S$.

Алгоритм реализации такой стратегии должен максимизировать сумму непосредственного дохода и значение функции подкрепления, уменьшенное коэффициентом дисконтирования:

$$a^* \leftarrow \arg \max_a [r(s, a) + \gamma \cdot V^*(\delta(s, a))]. \quad (3)$$

Уравнение (3) предполагает доступными сигнал подкрепления и функцию перехода. Однако в большинстве практических задач функция перехода недоступна. В работе [4] был впервые предложен одношаговый алгоритм Q -обучения, не использующий непосредственно функцию перехода. В этом алгоритме для определения оптимальной стратегии используется Q -функция, итеративную процедуру обновления которой можно представить в следующем виде:

$$Q_{t+1}(s, a) \leftarrow r + \gamma \cdot \max_{a \in A} Q(s', a), \quad (4)$$

$$Q_{t+1}(s, a) = Q_t(s, a) + \alpha \cdot (r + \gamma \cdot \max_{a' \in A} Q_t(s', a') - Q_t(s, a)), \quad (5)$$

где a – действие, вызывающее переход среды из состояния s в состояние s' ; α ($0 \leq \alpha \leq 1$) – коэффициент нормирования значений Q -функции.

В практических задачах алгоритм (4), (5) часто усиливают с помощью метода временной разности (Temporal Difference (TD)), что способствует ускорению обучения [5]. Идея TD-обучения состоит в использовании разности между двумя последовательными предсказаниями конечного ожидаемого результата вместо разности между текущим предсказанием и фактическим ожидаемым результатом.

Рассмотрим задачу применения TD-обучения для оценки функции (1), связанной с математическим ожиданием суммы будущих дисконтированных доходов:

$$V_t = E \left[\sum_{k=t}^{\infty} \gamma^{k-t} \cdot r_k \right], \quad \forall t. \quad (6)$$

Для получения разности между действительными доходами и их математическими ожиданиями будем использовать следующее обновляющее правило:

$$\Delta V_t = \alpha \cdot \left[\sum_{k=t}^{\infty} \gamma^{k-t} \cdot r_k - V_t \right], \quad (7)$$

где α – коэффициент обучения.

Оценку V_t можно осуществлять, используя аппроксиматор функции V^* с набором внутренних весов w . Эти весовые коэффициенты можно корректировать по методу градиентного спуска:

$$\Delta w_t = \eta \cdot \left[\sum_{k=t}^{\infty} [r_k + \gamma \cdot V_{k+1} - V_k] \cdot \gamma^{k-t} \right] \cdot \nabla_w \cdot V_t, \quad (8)$$

где η – коэффициент обучения ($\eta = \eta(\alpha)$).

Изменение общего веса за все время работы системы (обновляющее правило) определится следующей суммой:

$$\Delta w_t = [r_t + \gamma \cdot V_{t+1} - V_t] \cdot \sum_{k=0}^t \eta \cdot \gamma^{t-k} \cdot \nabla_w \cdot V_k, \quad (9)$$

где сумма с приращением обновляется на каждом шаге t .

В классическом методе Q -обучения Q -значения представляются таблицей с парами «состояние-действие». Практические задачи обычно характеризуются большим числом таких пар, что делает невозможным использование табличного представления Q -функций в большинстве случаев. Альтернативой таблицам перекодировки являются аппроксимирующие функции.

Рассмотрим возможность комбинирования Q -алгоритма с $TD(\lambda)$ -алгоритмом с целью повышения скорости процесса обучения. Отметим, что метод Q -обучения, основанный на обновлении (5), получает весьма ограниченную информацию от системы и выполняет модификации, основанные только на значениях смежных состояний. В $TD(\lambda)$ -алгоритме принимаются во внимание и предыдущие зафиксированные значения состояний. Заменяем Q -таблицу а дифференцируемой Q -функцией, имеющей некоторый набор внутренних параметров.

В соответствии с (5) TD -ошибка для Q -обучения представляется следующей зависимостью:

$$TD_{err} = Q_{t+1}(s, a) - Q_t(s, a) = (r + \gamma \cdot \max_{a' \in A} Q_t(s', a') - Q_t(s, a)). \quad (10)$$

Уравнение (10), определяющее ошибки временной разницы в терминах Q -значений, можно использовать как основу комбинированного Q/TD -обучения. Заменяя Q -значениями V -значения в уравнении (8), получаем:

$$\Delta w_t = \eta \cdot \left[r_t + \gamma \cdot \max_{a \in A} Q_{t+1} - Q_t \right] \cdot \nabla_w \cdot Q_t, \quad (11)$$

где $\nabla_w \cdot Q_t$ – градиент дифференцируемой Q -функции.

Отметим, что методы аппроксимации таблиц перекодировки, применяемые в комбинированном Q/TD -обучении должны обеспечивать возможность реализации алгоритмов в режиме on-line и иметь приемлемые потребности в памяти. Таким требованиям отвечают нейросетевые методы, в частности, методы, основанные на использовании многослойного персептрона (MLP)[5].

Рассмотрим многошаговый модифицированный алгоритм Q -обучения. Очевидно, что оценки Q -функций являются неточными, пока состояния финансового рынка не будут неоднократно наблюдаться. Последовательные уточнения динамики рынка могут быть учтены путем применения многошагового уточняющего правила следующего вида:

$$\Delta w_t = \eta \cdot \left[r_t + \gamma \cdot Q_{t+1} - Q_t \right] \cdot \sum_{k=0}^t (\gamma \cdot \lambda)^{t-k} \cdot \nabla_w \cdot Q_k. \quad (12)$$

Для аппроксимации таблиц перекодировки эффективным является применение специального метода коррекции весов MLP, именуемого методом «обратного переигрывания» (backward replay (BR))[6]. В соответствии с этим методом веса обновляются только при достижении системой поглощающих состояний. Использование BR-процедур предполагает необходимость хранения всех пар «состояние – действие», которые встречаются до достижения системой поглощающего состояния.

Альтернативой BR-методу может служить применение интерактивного обучения (IL) для ап-

проксимации Q -таблиц. Идея IL-метода состоит в обновлении аппроксимирующей функции на каждой итерации до достижения системой конечного состояния. Следует отметить, что рассмотренные основные и вспомогательные процедуры Q -обучения нашли практическое применение в основном для поиска управляющих стратегий в технических системах (в частности, для управления роботом). Использование методов Q -обучения в трейдинговых системах имеет свои специфические особенности.

Для работы описанного Q/TD -алгоритма на электронной бирже в составе трейдинговой системы необходимо в реальном времени получать значения доступных индикаторов, помогающих обнаружить тенденции изменения основных показателей биржевого рынка. Индикаторы можно разделить на две большие группы: трендовые индикаторы и осцилляторы. Трендовые индикаторы применяются при анализе трендовых рынков и неэффективны, когда тренд отсутствует. Осцилляторы, наоборот, плохо работают на трендовых рынках и хорошо, когда тренд отсутствует. Ниже приведен пример формирования оптимальных стратегий с использованием алгоритма Q -обучения, иллюстрирующих принцип работы трейдерной системы, основанной на получении сигналов подкрепления от финансового рынка.

3. Пример Q -обучения на финансовом рынке

На рис. 1 рассматривается упрощенная торговая ситуация, где трейдер имеет два индикатора финансового рынка и по сигналам этих индикаторов он должен принимать решения. В этой упрощенной ситуации предполагается, что каждый индикатор может советовать совершить либо покупку (buy), либо продажу (sell). Индикаторы нейтральной позиции, советующие отказаться от торговых транзакций, здесь отсутствуют. Так как два индикатора могут принимать лишь по два значения каждый, то финансовый рынок (среда) может иметь 4 возможных состояния (state). В любом из этих состояний среды трейдер может принять два решения - купить или продать. Очевидно, что набор действий, доступных в каждом из состояний, может быть представлен следующим образом:

$$A(\text{state } i) = \{buy, sell\}, i = 1, 2, 3, 4.$$

Любое действие, осуществленное трейдером, приводит к переходу от одного состояния окружающей среды к другому в соответствии с некоторым набором вероятностей, определенным рыночной ситуацией.

Состояние окружающей среды характеризуется значениями индикаторов, и в любом состоянии трейдер должен выбрать действие «buy» или действие «sell». Для понимания процедуры обучения,

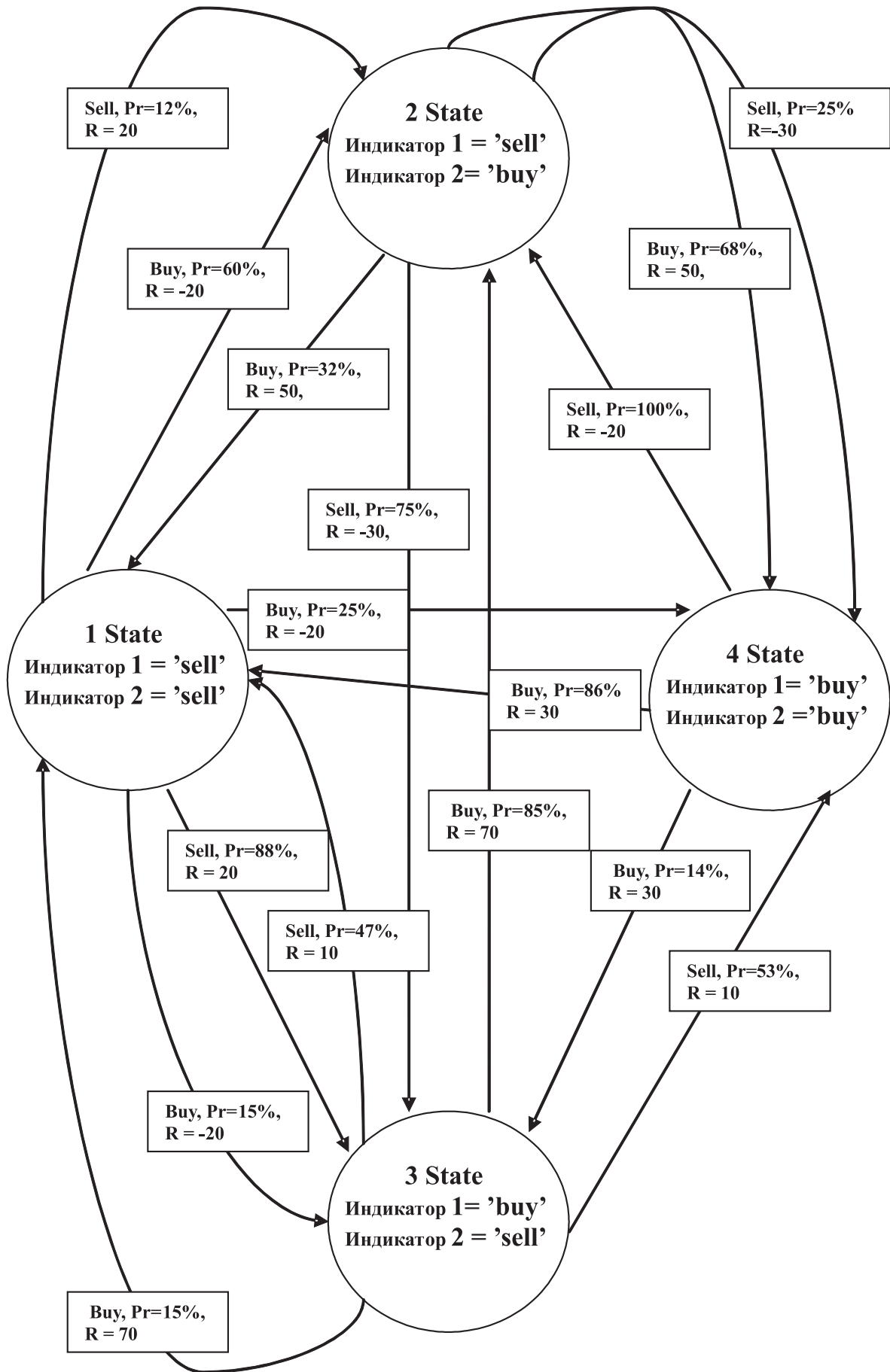


Рис. 1. Пример Q-обучения на финансовом рынке

представленной на рис. 1, рассмотрим, например, прямоугольник в левом верхнем углу этого рисунка, содержащий данные: «sell, Pr = 12%, R = 20». Это означает, что здесь переход из состояния 1 в состояние 2 будет иметь место, если трейдер в состоянии 1 выберет действие «sell», и что это решение приведет к переходу в состояние 2 с вероятностью 0,12, при котором трейдер получит сигнал подкрепления R = 20. Отметим, что решение «sell» в состоянии 1 может также привести к переходу в состояние 3 с вероятностью 0,88. Суммарная вероятность этих двух переходов равна 1, что указывает на нулевую вероятность перехода в состояние 4 при решении «sell» в состоянии 1.

Значения сигналов подкреплений и вероятностей переходов на рис.1 были выбраны произвольно, поскольку рассматриваемый пример имеет лишь иллюстративный характер. На практике эти значения изменяются в соответствии с динамической природой рынка, но на рис.1 они зафиксированы для понимания ключевых особенностей Q-обучения. Предположим, что начальные значения $Q_0(s,a)$ являются нулевыми и что коэффициент дисконтирования $\gamma = 0,85$. Допустим также, что имеет место следующая последовательность «состояние – действие»:

(state1, sell) → (state3, buy) → (state3, buy) →
 (state2, sell) → (state3, sell) → (state4, buy) →
 (state3, buy) → (state2, buy) → (state1, buy) →
 (state2, sell) → (state3, ...) → ...

Рассмотрим для такой последовательности процедуру пересчета Q-значений по описанному выше методу Q-обучения. Отметим, что здесь каждый шаг Q-обучения осуществляется по следующей рекуррентной зависимости:

$$Q_{t+1}(s,a) = Q_t(s,a) + \alpha \cdot (r + \gamma \cdot \max_{a' \in A} Q_t(s',a') - Q_t(s,a)),$$

где α – коэффициент обучения, принятый здесь равным 0,15.

Последовательность первых 4 шагов модификаций имеет вид:

1) (state1, sell, 20, state3)

$$Q_1(\text{state 1, sell}) = Q_0(\text{state 1, sell}) + 0,15 \left[20 + 0,85 \max_a Q_0(\text{state 3, } a) - Q_0(\text{state 1, sell}) \right] = -0 + 0,15 \left[20 + 0,85 \max\{0,0\} - 0 \right] = 0 + 0,15 \cdot 20 = 3;$$

2) (state3, buy, 70, state2)

$$Q_2(\text{state 3, buy}) = Q_1(\text{state 3, buy}) + 0,15 \left[70 + 0,85 \max_a Q_1(\text{state 2, } a) - Q_1(\text{state 3, buy}) \right] = 0 + 0,15 \left[70 + 0,85 \max\{0,0\} - 0 \right] = 0 + 0,15 \cdot 70 = 10,5;$$

3) (state2, sell, -30, state3)

$$Q_3(\text{state 2, sell}) = Q_2(\text{state 2, sell}) + 0,15 \left[-30 + 0,85 \max_a Q_2(\text{state 3, } a) - Q_2(\text{state 2, sell}) \right] = 0 + 0,15 \left[-30 + 0,85 \max\{10,5,0\} - 0 \right] = 0 + 0,15 \cdot (-30 + 0,85 \cdot 10,5) = -3,16;$$

4) (state3, sell, 10, state4)

$$Q_4(\text{state 3, sell}) = Q_3(\text{state 3, sell}) + 0,15 \left[10 + 0,85 \max_a Q_3(\text{state 4, } a) - Q_3(\text{state 3, sell}) \right] = 0 + 0,15 \left[10 + 0,85 \max\{0,0\} - 0 \right] = 0 + 0,15 \cdot 10 = 1,5.$$

Ниже приведена Q-таблица, сформированная после четвертого шага.

Таблица 1

Q-таблица после 4 шагов обучения

$Q_4(s,a)$	Buy	Sell
state 1	0	3
state 2	0	-3,16
state 3	10,5	1,5
state 4	0	0

Последовательность следующих пяти шагов пересчета Q-значений имеет вид:

(state4, buy) → (state3, buy) → (state2, buy) →
 (state1, buy) → (state2, sell) → (state3, ...) → ...

Q-таблица, сформированная после девятого шага, приведена ниже.

Таблица 2

Q-таблица после 9 шагов обучения

$Q_9(s,a)$	Buy	Sell
state 1	0,55	3
state 2	7,88	-4,71
state 3	19,43	1,5
state 4	5,84	0

Стратегия Q-обучения после 9 шагов определяется следующим уравнением:

$$\pi^*(s) = \arg \max_a Q_9(s,a).$$

Определим последовательность действий трейдера в соответствии с этой стратегией:

- если система находится в состоянии «state1», то выбирается действие «sell», так как $3 > 0,55$;
- если система находится в состоянии «state2», то выбирается действие «buy», так как $7,88 > -4,71$;
- если система находится в состоянии «state3», то выбирается действие «buy», так как $19,43 > 1,5$;
- если система находится в состоянии «state4», то выбирается действие «buy», так как $5,84 > 0$.

В статической окружающей среде, рассматриваемой в этом примере, Q -значения в конечном счете сошлись бы к фиксированным значениям и, таким образом, трейдер мог бы наиболее эффективно использовать для торговли значения Q -таблиц только после их стабилизации. В реальных трейдерских системах, когда окружающая среда является динамической, трейдеру, кроме анализа сходимости Q -значений, приходится использовать некоторые дополнительные критерии, чтобы определить наиболее целесообразный момент использования значений Q -таблиц для торговли.

Выводы

Результаты, полученные при моделировании торговых стратегий по данным валютного рынка FX, подтверждают работоспособность и перспективность применения машинного обучения с подкреплением в трейдинговых системах. Практическая реализация принципов электронной торговли с использованием рассмотренных выше процедур, осуществленная авторами настоящей статьи, состоит в следующем:

- составлен банк Q -обучающих алгоритмов с возможностью программного его пополнения пользователем;

- предложен вычислительный алгоритм, основанный на использовании обучающей модели с RL, и дополнительные средства, позволяющие принимать оперативные решения относительно входа в рынок и выхода из рынка;

- разработан программный модуль Q -Trader, основанный на алгоритме, который позволяет принимать действия, имитирующие процесс торговли. Модуль может соединяться с сервером брокера, постоянно загружать данные с сервера, анализировать ситуацию на рынке, а затем обеспечивать формирование рекомендаций по ведению торговли.

Список литературы: 1. Люггер, Д. Искусственный интеллект: стратегии и методы решения сложных проблем [Текст] / Д.Люггер // М.: Издательский дом «Вильямс».

– 2003. – 864 с. 2. “Computational Learning Techniques for Intraday FX Trading Using Popular Technical Indicators / M. Dempster, T. Payne, Y Romahi, G. Thompson // IEEE Transactions on Neural Networks, Vol. 12, 4, July 2001. – P. 744-754. 3. Dempster, M. Intraday FX trading: An evolutionary reinforcement learning approach. Intelligent data engineering and automated learning / M. Dempster, Y. Romahi // Proceedings of the IDEAL 2002 International Conference. – 2002. – P. 347-358. 4. Sutton, R. Learning to Predict by the Methods of Temporal Differences, Machine Learning, 3, pp. 9-44, 1998. 5. Барский, А.Б. Нейронные сети: распознавание, обучение, принятие решений [Текст] / А.Б. Барский. – М.: Финансы и статистика, 2004. – 320 с. 6. Hryshko A. An Implementation of Genetic Algorithms as a Basis for a Trading System on the Foreign Exchange Market / A. Hryshko, T. Downs // Proceedings of the 2003 Congress on Evolutionary Computation. – 2003. – P. 1695-1701.

Поступила в редколлегию 19.02.2010 г.

УДК 519.62

Визначення стратегій в комп'ютерних трейдингових системах на основі методів машинного навчання / А.О. Гришко, С.Г. Удовенко, Л.Е. Чала // Біоніка інтелекту: наук.-техн. журнал. – 2010. – № 1 (72). – С. 18–23.

У статті розглядаються стратегії трейдера у системах біржової торгівлі, що використовують методи машинного навчання з підкріпленням. Запропоновані алгоритми дозволяють перевищити за своїми характеристиками інші алгоритми машинного навчання, які використовуються на фінансових ринках. Розглянуті принципи трейдингової торгівлі засновані на розроблених методах. Методи реалізовано програмно та протестовано.

Табл. 2. Іл. 1. Бібліогр.: 6 найм.

УДК 519.62

Strategies of computing markets systems based on machine learning methods / A.A. Hryshko, S.G. Udoenko, L.E. Chalya // Bionics of Intelligence: Sci. Mag. – 2010. – № 1 (72). – P. 18–23.

This paper is devoted to application of machine learning for computing markets systems. The algorithms developed in the proposed learning system is shown to outperform other machine learning algorithms used on financial markets. The principles of trading markets based on the development of novel trading strategies are considered. The new methods are implemented in software. The results of testing of methods are presented.

Tab. 2. Fig. 1. Ref.: 6 items.

УДК 007.5



ЛОГИЧЕСКАЯ МОДЕЛЬ ПРЕДСТАВЛЕНИЯ БИЗНЕС-ПРОЦЕССА ДЛЯ РЕШЕНИЯ ЗАДАЧ ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА ПРОЦЕССОВ

В.М. Левыкин¹, С.Ф. Чалый², А.А. Кривчикова³, С.Н. Селезнев⁴

¹ ХНУРЭ, г. Харьков, Украина, levykin@kture.kharkov.ua

² ХНУРЭ, г. Харьков, Украина, s_chaliy@mail.ru

³ ХНУРЭ, г. Харьков, Украина, krivchikovaanna@gmail.com

⁴ ХНУРЭ, г. Харьков, Украина, rp01@mail.ru

В работе предложен и обоснован логический подход к формализации многоуровневого представления бизнес-процессов (БП), ориентированный на решение задач интеллектуального анализа БП (process mining). Получена логическая модель представления бизнес-процесса, которая обеспечивает возможность решения задач управления БП и создает возможности для реализации формализованного подхода к формированию и верификации описания бизнес-процессов на основе интеллектуального анализа процессов.

БИЗНЕС-ПРОЦЕССЫ, ПРАВИЛА, ОБЪЕКТЫ, ПРОЦЕДУРЫ

Введение

Современный этап процессного управления организациями характеризуется необходимостью построения гибких бизнес-процессов (БП), ориентированных на пользователя и изменяющихся на основе знаний. Традиционно в основе процессного подхода к управлению организацией лежит выделение в организации бизнес-процессов и управление этими процессами. При этом под БП обычно понимают устойчивую, целенаправленную совокупность взаимосвязанных видов деятельности (последовательности работ), которая по определенной технологии преобразует входы в выходы, представляющие ценность для потребителя [1]. Бизнес-процессы имеют сложную многоуровневую структуру и включают в себя, в частности, следующие ключевые элементы: ресурсы; процедуры процесса в заданной последовательности; знания о взаимосвязи процедур, исполнителях, поставщиках, потребителях, выраженные в форме правил.

В последние годы произошли значительные изменения в области управления бизнес-процессами, направленные на унификацию существующих подходов. Вместо проектирования специализированных информационных систем для каждого предприятия теперь разрабатываются процессно-ориентированные системы, направленные на поддержку ряда видов деятельности на основе построения формальной процессной модели. Иными словами, основной акцент в данной области смещается с непосредственно программирования в сторону построения максимально точной и детальной процессной модели, которая реализуется в рамках системы управления бизнес-процессами (СУБП) [2]. Жизненный цикл такой системы включает в себя этапы [3] формализации описания процесса; конфигурирования информационной системы; выполнения процесса; анализа результатов выполнения с целью улучшения процесса. На этапе вы-

полнения формируется журнал событий, который отражает текущее поведение БП. Данный журнал используется на этапе анализа для выбора направления дальнейшей реорганизации и улучшения каждого процесса и системы в целом. Таким образом, рассмотренный жизненный цикл ориентирован на непрерывное совершенствование систем управления бизнес-процессами.

Следует отметить, что среди рассмотренных четырех этапов существующие СУБП позволяют автоматизировать лишь этап выполнения процесса. Остальные этапы ориентированы на интеллектуальную деятельность человека и требуют применения знаний для анализа и построения улучшенной модели процесса, а также усовершенствованной версии системы управления. Следовательно, возникает проблема разработки методов и моделей анализа, построения и усовершенствования БП на основе знаний о порядке протекания и ограничениях на выполнение процессов. Указанные вопросы рассматриваются в научном направлении интеллектуального анализа процессов (process mining). Данное направление ориентировано на построение систем, моделирующих интеллектуальную деятельность человека по нахождению новых и доступных для интерпретации знаний, необходимых для выявления взаимосвязей между элементами бизнес-процессов, ограничений и условий их выполнения, а также восстановления структуры процесса [2]. В качестве исходной информации для process mining используются зафиксированные во времени данные о результатах функционирования существующих процессов, а также документальные требования к таким БП. Решение указанной проблемы основывается на обобщенном формализованном описании всех элементов бизнес-процесса, что требует разработки модели представления БП, обеспечивающей возможность его реорганизации на основе знаний.

1. Постановка задачи

Традиционное представление БП ориентировано на формализацию последовательности процедур между входом и выходом процесса, а также взаимосвязей между ними. В то же время, для задач интеллектуального анализа процессов такое представление должно отражать его многоуровневую структуру, основанную на знаниях о процессе, ограничениях на БП и порядке его реорганизации. В работе [4] предложена обобщенная многокомпонентная модель БП, которая учитывает многоуровневую структуру процесса и отражает элементы знаний в виде правил верхнего уровня модели (рис. 1).

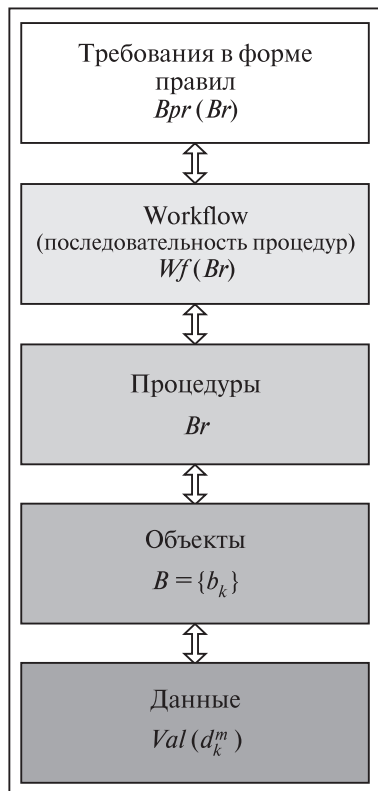


Рис. 1. Многокомпонентная модель представления бизнес-процессов

Здесь $Br = \{Br_j\}$ – это множество бизнес-процедур; $Wf(Br)$ – последовательность выполнения процедур Br_j ; $Vpr(Br)$ – правила выполнения БП, которые соответствуют функциональным требованиям к бизнес-процессам; $B = \{b_k\}$ – набор бизнес-объектов бизнес-процесса; $Val(b_k^m)$ – текущие значения данных – атрибутов объектов b_k [4]. Отметим, что многокомпонентная модель носит преимущественно декларативный характер и требует дальнейшего развития с учетом рассмотренного жизненного цикла системы управления бизнес-процессами. Соответственно, сформулируем требования к модели представления бизнес-процесса для решения задачи интеллектуального анализа данных:

- формализация основных элементов БП: знаний о порядке его выполнения; процедур; объектов, с которыми оперирует бизнес-процесс; исполнителей;

- возможность непосредственной интерпретации модели (ее исполнения);

- единая форма представления для всех элементов БП, включая знания о процессе, а также для взаимосвязей между элементами;

- возможность формализованного изменения последовательностью процедур БП, исполнителей процедур на основе результатов process mining.

Сформулированные требования отражают специфику как бизнес-процессов, так и интеллектуального анализа процессов и в значительной степени определяют выбор математического аппарата логики для построения модели представления БП [5].

В соответствии с изложенным, сформулируем задачу построения логической модели представления бизнес-процесса.

Исходными данными задачи являются:

- 1) множество элементов бизнес-процесса, включая знания, процедуры, объекты и исполнителей;

- 2) многокомпонентная модель представления БП, которая отражает иерархию взаимосвязей между элементами процесса.

Требуется получить логическую модель представления бизнес-процесса, которая удовлетворяла бы изложенным требованиям и позволяла бы выполнить оценку процесса (в частности, оценку достижимости его конечного состояния).

Для решения поставленной задачи необходимо выполнить следующие шаги:

- 1) провести анализ существующих подходов к описанию элементов бизнес-процесса и взаимосвязей между этими элементами;

- 2) обосновать выбор математического аппарата;

- 3) адаптировать выбранный математический аппарат с учетом особенности знаний ориентированной структуры бизнес-процесса;

- 4) построить логическую модель многоуровневого представления бизнес-процесса.

2. Анализ существующих подходов к описанию и моделированию бизнес-процессов

В настоящее время разработан целый ряд подходов к моделированию бизнес-процессов, ориентированных на специфику различных отраслей. Указанные подходы целесообразно разбить на три группы по признакам учета взаимосвязей процедур, ограничений на процесс и учета событий, происходящих при выполнении БП:

- 1) граф управления потоком работ;

- 2) темпоральные ограничения;

- 3) правила вида «событие-условие-действие».

Граф управления потоком работ позволяет визуально отобразить зависимости между процедурами процесса и поэтому часто применяется в коммерческих системах управления бизнес-процессами.

В соответствии с данным подходом реализованы следующие технологии: EPC (event-driven process chain) – технология описания процессов на основе цепочки событий; ERM (entity relationship model) – модель сущность-связь для описания структуры данных; BPMN (business process modeling notation) – технология графического представления бизнес-процессов.

При реализации рассмотренных подходов используются различные способы формализации, в частности аппарат расширенных сетей Петри. В данном случае БП представляется сетью Петри. Каждый этап выполнения процесса (либо процедура БП) отображается переходом сети Петри. Передача управления между процедурами формализуется с помощью позиций сети Петри. Информация, используемая при выполнении бизнес-процесса и отражающая свойства объектов процесса (бизнес-объектов), отображается маркерами сети.

Следует отметить, что сети Петри, представляющие описание бизнес-процесса в workflow-системе, обычно выделяют в отдельный класс сетей Петри, именуемый WPN (Workflow Petri Nets). Такое выделение класса сетей связано с ограничениями процесса на последовательность процедур, что и обуславливает дополнительные свойства WPN. Объектно-ориентированные расширения сетей Петри позволяют интегрировать возможности workflow и UML для описания, анализа, моделирования и управления бизнес-процессами современного предприятия с использованием открытых стандартов представления информации [1].

Подход на основе темпоральных (временных) ограничений представляет возможности для моделирования графа управления потоком работ без дуг с помеченными условиями перехода. Данный подход имеет ограниченные возможности и не позволяет естественным образом представить коррекцию потока работ, а также отразить поддержку принятия решений и верификацию целостности бизнес-процессов. Подход, основанный на правилах вида «событие-условие-действие», ориентирован на построение бизнес-процессов с изменяемой структурой и позволяет отобразить знания о функционировании и взаимосвязи процедур процесса. В то же время, в соответствии с данным подходом не всегда видна целостная структура БП, поэтому представляется целесообразным его сочетание с графом управления потоком работ.

Следует отметить, что бизнес-процессы предприятия могут быть описаны с помощью стандартных блок-схем. Они содержат некоторые дополнительные специальные графические объекты,

использование которых позволяет делать блок-схемы процессов более наглядными и понятными для исполнителей [1].

Таким образом, рассмотренные подходы ориентированы преимущественно на построение фиксированной структуры процесса как последовательности взаимосвязанных процедур, выполнение которых ограничено причинно-следственными связями и временными рамками. В то же время решение задач интеллектуального анализа процессов основано на использовании знаний, что требует включения формализованного их представления в структуре БП.

3. Обоснование логического подхода к формализации бизнес-процессов

На рис. 1 была приведена структура многокомпонентной модели представления бизнес-процесса. Такая модель позволяет описать реализацию БП в виде графа с конечным набором начальных состояний. Указанный граф может быть представлен в виде логических формул, отражающих состояния вершин и правила перехода между вершинами. Отметим, что временные параметры бизнес-процесса требуют применения логического аппарата, который бы учитывал фактор времени. Исходя из изложенного, используем для описания БП аппарат темпоральной модальной логики STL*. В темпоральной логике синтаксис операторов времени позволяет корректно описать временные свойства процесса. Для того чтобы избежать неоднозначности в описании потоков работ, используются логические и модальные операторы. Логические операторы связи позволяют описать ветвление и слияние процесса [6].

Можем формализовать модель описания бизнес-процессов как структуру ветвящегося времени $M = (S, S_0, R, L)$, где:

1. S – конечное множество состояний;
2. S_0 – множество начальных состояний;
3. $R \subseteq S \times S$ – отношение переходов;
4. L – функция истинности.

Заметим, что M является ориентированным графом с набором вершин S , с набором дуг R и с метками узлов, которые задаются функцией L . Синтаксисом темпоральной логики определяются правила составления формальных утверждений. Синтаксис STL* включает в себя следующие наборы кванторов пути: \forall – квантор всеобщности, указывающий на то, что данное свойство выполнено для всех путей реализации бизнес-процесса; \exists – квантор существования, указывающий на то, что данное свойство выполнено для некоторого пути реализации БП.

В логике STL* выделяют два типа формул: формулы состояний и формулы пути. Формулы состояний описывают свойства соответствующе-

го состояния, формулы пути описывают свойства соответствующего пути. Приведем синтаксис формул состояния и синтаксис формул пути.

1. Синтаксис формул состояний. Пусть AP – множество атомарных высказываний.

- Если $f \in AP$, то f – это формула состояний.
- Если f и g формулы состояний, то $\neg f, f \wedge g, f \vee g$ – формулы состояний.
- Если f – формула пути, то $\forall f, \exists f$ – формула состояния.

2. Синтаксис формул пути.

- Если f – формула состояний, то f – формула пути.
- Если f и g формулы пути, то $\neg f, f \wedge g, f \vee g$ – формулы пути.

Отметим, что составляющие многокомпонентной модели представления БП, такие как требования в форме правил, последовательность процедур (Workflow), процедуры, объекты и данные являются конечным набором состояний и могут быть описаны с помощью формул состояний СТЛ* логики. Исходя из рис. 1 и применяя синтаксис СТЛ* логики, взаимосвязи между знаниями о предметной области, выраженными в форме правил Bpr и последовательностью процедур Br_j бизнес-процесса, выражаются логическим произведением правил Bpr_j и соответствующих процедур Br_j : $Bpr_j \wedge Br_i \rightarrow Bpr_l, j=1, \overline{J}, i=1, \overline{I}, l=1, \overline{L}$. Иными словами, на основе логического произведения процедур и правил мы можем упорядочить знания о структурных взаимосвязях бизнес-процесса в виде иерархии бизнес-правил и таким образом создать возможность оценки достижимости конечного результата(состояния) процесса при имеющихся знаниях и ограничениях в предметной области.

Аналогичным образом логическое умножение процедур $Br_j, j=1, \dots, n$ и объектов процедур $b_k, k=1, \dots, n$: $Br_j \wedge b_k$ позволяет оценить полноту процесса путем определения тех объектов БП, которые не обрабатываются процедурами.

Таким образом, рассмотренный математический аппарат позволяет удовлетворить требованиям о представлении бизнес-процесса для решения задачи интеллектуального анализа процессов.

4. Логическая модель многоуровневой представления бизнес-процессов

В соответствии с рассмотренным подходом усовершенствуем представленную на рис. 1 многокомпонентную модель. Предлагаемая логическая модель представления БП дополнительно учитывает: роли исполнителей; структуру объектов, с которыми оперирует бизнес-процесс; знания о функционировании БП (рис. 2).

Предлагаемая модель содержит знания в форме правил $Bpr = \{Bpr_j\}$ двух видов: правила, отражающие порядок выполнения процедур БП, взаимо-

связи между процедурами, а также ограничения на последовательность выполнения процедур; правила, отражающие порядок обработки объектов бизнес-процесса и порядок обработки объектов.

Усовершенствованная модель представления бизнес-процессов включает в себя роли исполнителя процедур. Обозначим Rl_j роль исполнителя Isp_i процедуры. Процедуры Br_j объединяются в Workflow на основе рассмотренных ранее правил. С каждой процедурой связаны исполнители посредством механизма ролей. Совокупность объектов бизнес-процесса, их атрибутов, данных, а также соответствующих правил обработки составляют ресурсы процесса. Правила обработки объектов в предлагаемой модели позволяют структурировать и формализовать нижний уровень бизнес-процесса. Такие правила позволяют выделить ключевые объекты, оценить эффективность их обработки и тем самым создать условия для усовершенствования БП [7].

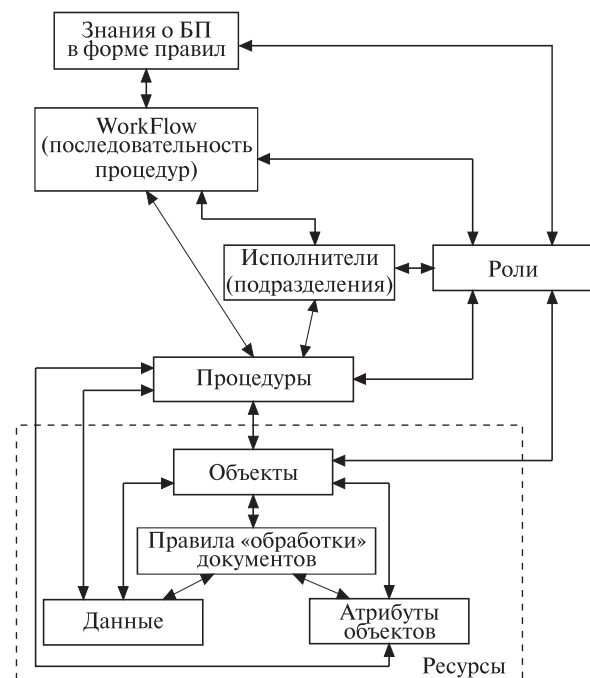


Рис. 2. Многоуровневая модель представления бизнес-процесса с изменяющейся структурой

В соответствии с предлагаемым подходом, взаимосвязь бизнес-процедуры Br_j и роли исполнителя этой процедуры Rl_j естественным образом представляется в виде конъюнкции $Br_j \wedge Rl_j, j=1, \overline{J}$ и позволяет оценить наличие исполнителей Isp_i соответствующей процедуры моделируемого процесса.

Аналогично, используя синтаксис формулы состояния, можем применить конъюнкцию для i -го исполнителя j -ой роли: $Rl_j \wedge Isp_i \rightarrow Isp_j, i=1, \overline{I}, j=1, \overline{J}$. Таким же образом конъюнкция используется для реализации правила-ограничения Bpr_j на j -ю роль бизнес-процедуры: $Bpr_i \wedge Rl_j \rightarrow Bpr_j$.

Адаптируем интерпретацию модальных операторов логики [6] с учетом особенностей предметной области: FBr_j — «в конечном итоге должна выполняться процедура Br_j »; GBr_j — «всегда выполняется процедура Br_j »; XBr_j — «в следующий момент времени будет выполняться процедура Br_j »; $Br_i \cup Br_j$ — «Будет выполняться процедура Br_i до тех пор, пока не станет выполняться процедура Br_j », при $i \neq j$. Указанные операторы позволяют формализовать workflow-уровень модели. Для логической связи процедур в workflow — последовательности используем импликацию (\Rightarrow) в следующем смысле «Если процедура Br_i выполнена, то процедура Br_j также выполнена», и эквиваленцию (\Leftrightarrow), в смысле «процедура Br_i истинна тогда и только тогда, когда истинна процедура Br_j » [6]. Аналогично традиционной логике: $Br_i \vee Br_j \Leftrightarrow \neg(\neg Br_i \wedge \neg Br_j)$, а также $Br_i \Rightarrow Br_j$ то же самое, что $\neg Br_i \vee Br_j$. Также, $Br_i \vee \neg Br_i$ означает, что бизнес-процедура Br_i выполнена или не выполнена. Если связь между состояниями прямая, то используется оператор истины (true), если не прямая, тогда ложь (false) [6].

Таким образом, предложенная логическая модель позволяет сочетать декларативное многоуровневое представление бизнес-процессов с логическим описанием последовательности процедур и обеспечивает использование знаний в виде правил.

Выводы

В работе предложен и обоснован логический подход к формализации многоуровневого представления бизнес-процессов, ориентированный на решение задач интеллектуального анализа БП (process mining). Предложенный подход основан на представлении в виде логических формул средствами СТЛ* логики как собственно модели бизнес-процесса, так и знаний о порядке выполнения процедур БП, порядке обработки объектов бизнес-процесса, взаимосвязи между процедурами. Данный подход создает условия для реализации жизненного цикла БП и непрерывного их улучшения с учетом сравнительной оценки стоимости и времени выполнения различных вариантов реализации процессов.

Полученная логическая модель представления бизнес-процесса отличается от существующих единым формальным описанием различных уровней БП, порядком выполнения процедур процесса, а также знаний об их функционировании.

С одной стороны, предложенная модель обеспечивает возможность решения ряда задач управления БП: анализ и оптимизация бизнес-процесса по заданным показателям эффективности; формирование эффективной системы управления, основанной на процессном подходе; разработка

нормативных документов, регламентирующих технологию выполнения бизнес-процесса (либо его отдельных элементов); подготовка к созданию и внедрению системы управления бизнес-процессами.

С другой стороны, разработанная модель создает возможности для реализации формализованного подхода к формированию и верификации описания бизнес-процессов на основе интеллектуального анализа процессов.

Список литературы: 1. Репин, В.В. Процессный подход к управлению Моделирование бизнес-процессов [Текст]: учеб./ В.В. Репин В.Г., Елиферов. — М.: РИА «Стандарты и качество», 2004. — 408 с. ISBN 5-94938-018-5. 2. Trends in Business Process Analysis: From Verification to Process Mining [Text]: Proceedings of the 9th International Conference on Enterprise Information Systems, ICEIS, 2007. Medeira/editors: Cordeiro J. Cardoso and J. Filipe. — Institute for Systems and Technologies of Information, Control and Communication, INSTICC, Portugal, 2007. — pages 12–22. 3. Елиферов, В.Г. Бизнес-процессы: регламентация и управление [Текст]: учеб./ В.Г.Елиферов, В.В. Репин— М.:ИНФРА-М, 2005. — 319 с. 4. Левыкин, В.М. Формализованное описание бизнес-процессов в АИС [Текст] / В.М.Левыкин, С.Ф. Чалый //Штучний інтелект. — 2003. — № 2. — С. 28–34. 5. Калянов, Г.Н. Теория и практика реорганизации бизнес-процессов [Текст]: учеб./ Г.Н.Калянов. — М.: СИНТЕГ, 2000. —212 с. 6. Allen Emerson E. Temporal and modal logic [Text]:/ Allen Emerson. Computer Sciences Department University of Texas, Austin, 1995. — 68 p. 7. Левыкин, В.М. Модели представления знаний для бизнес-процессов с изменяемой структурой [Текст] / В.М.Левыкин, С.Ф. Чалый // Бионика интеллекта. — 2008. — № 1(68). — С.40–44.

Поступила в редколлегию 19.02.2010 г.

УДК 007.5

Логічна модель представлення бізнес-процесу для вирішення задач інтелектуального аналізу процесів / В.М.Левікін, С.Ф.Чалий, Г.О. Кривчікова, С.М. Селезньов // Біоніка інтелекту: наук.-техн. журнал. — 2010. — № 1 (72). — С. 24–28.

В статті обговорюється підхід до формалізації багаторівневого представлення бізнес-процесів. Отримана модель забезпечує можливість вирішення задач управління бізнес-процесів. Використання запропонованих правил створює умови для виконання реорганізації бізнес-процесів із змінною структурою.

Л. 2. Бібліогр.: 7 найм.

UDC 007.5

Logical model business process representation for solving problems of intellectual analysis process / V.M. Levikin, S.F. Chaliy, G.O. Kryvchikova, S.M. Selesniyov // Bionics of Intelligence: Sci. Mag. — 2010. — № 1 (72). — P. 24–28.

An approach to formalization of multi-representation of business processes is discussed. The resulting model enables solving the management of business processes. The use of the offered rules creates terms for implementation of reengineering of business-processes with a variable structure.

Fig. 2. Ref.: 7 items.

УДК519.85



РОЗМІЩЕННЯ ПРЯМОКУТНИХ ГРАФІЧНИХ ЕЛЕМЕНТІВ В ПОЛІГРАФІЧНОМУ ВИРОБНИЦТВІ

О. В. Баранов¹, Д.В. Грищай², І. В. Гребеннік³, О.В. Панкратов⁴, А.М. Чугай⁵

¹ ХНУРЕ, м. Харків, Україна, aleksey.baranov@gmail.com

² ХНУРЕ, м. Харків, Україна, gdv_denys@yahoo.com

³ ХНУРЕ, м. Харків, Україна, stdep@kture.kharkov.ua

⁵ ІПМАШ НАНУ, м. Харків, Україна, pankratov@mail.ru

⁶ ІПМАШ НАНУ, м. Харків, Україна, chugay@ipmach.kharkov.ua

Аналізується проблема розкрою матеріалів при виробництві поліграфічної продукції, описуються її особливості. Будується математична модель розміщення множини прямокутних графічних елементів при розкрої матеріалу для видавничо-поліграфічного виробництва. Проводяться обчислювальні експерименти та аналізуються їх результати.

МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ, ГЕОМЕТРИЧНЕ ПРОЕКТУВАННЯ, УПАКОВКА, ОПТИМІЗАЦІЯ, Ф-ФУНКЦІЯ, NP-СКЛАДНА ЗАДАЧА

Вступ

В процесі підготовки книжково-журнальних видань виникає ряд задач, пов'язаних із розміщенням (*nesting problems*) [1, 2]. Як приклад, можна навести розміщення рекламних блоків, які мають фіксований набір розмірів, але розміщуватися можуть у різні формати паперу; розміщення наліпок на окремі аркуші паперу, які доповнюють видання та надають йому більш цікавий вигляд, тощо. Взагалі графічні елементи можуть мати різні розміри, кількість та форми: прямокутну (*rectangular shaped objects*) або вільну (*irregular shaped objects*). Розміщення може враховувати ортогональну орієнтацію об'єктів, які розміщуються. Також слід враховувати технологічні вимоги, серед яких є розміщення із заданою відстанню між об'єктами, розміщення з урахуванням довжини ножів, розміщення об'єктів по групах, тощо. Все це впливає на складність розв'язання отриманої задачі розміщення [3, 4].

Метою роботи є побудова математичних моделей та методів розв'язання деяких задач розміщення прямокутних графічних елементів із заданою відстанню між елементами та врахуванням можливості повороту об'єктів.

1. Постановка задачі

Розглянемо постановку задачі. Нехай є множина графічних елементів, які необхідно розмістити в задану прямокутну область. Відстань між розміщуваними елементами регламентується технологічними вимогами до виготовлення штанц-форм, тобто відома заздалегідь, тому можна вважати, що цю відстань враховано у розмірах самих елементів (рис. 1).

Область розміщення завжди є прямокутною. Розмір області визначається форматом паперу, який використовується при друкуванні видання [2]. Кількість аркушів, які використовуються при формуванні області розміщення, визначається

ідеєю та собівартістю проекту, тобто у багатьох випадках відома до початку розміщення. Приклади областей розміщення наведені на рис. 2 у вигляді сірих прямокутників.

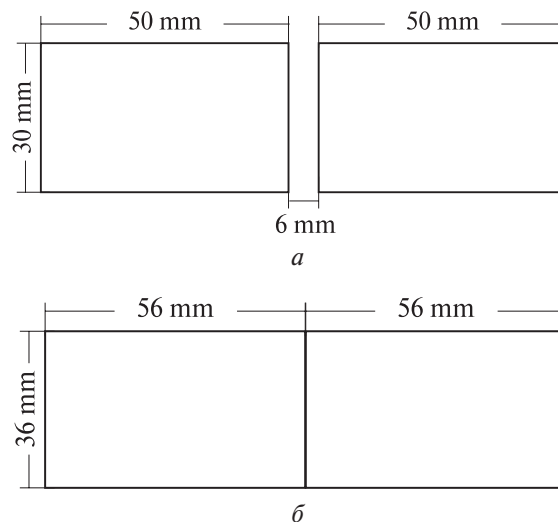


Рис. 1. Врахування відстані між елементами на етапі формування розмірів елементів (*a* — розміри елементів та відстань між ними, *b* — розміри елементів після врахування відстані між ними)

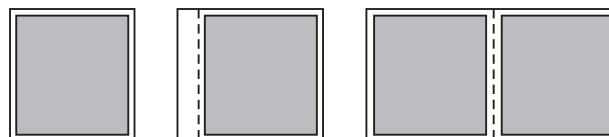


Рис. 2. Области розміщення графічних елементів

Розміщення графічних елементів може мати такі особливості.

1. Необхідно розмістити всю множину графічних елементів таким чином, щоб кількість аркушів була якомога менша.

2. Необхідно розмістити якнайбільше графічних елементів, при цьому врахувати, що кількість аркушів повинна бути не більша за задану.

2. Математична модель задачі

Маємо N прямокутних елементів $P_i \subset R^2$ вигляду $P_i = \{x \in R^2 : x = (x_1, x_2) | 0 \leq x_j \leq a_{ij}, j \in \{1, 2\}\}$, де $x = (x_1, x_2)$ — полюс, $a_i = (a_{i1}, a_{i2})$ — вектор метричних характеристик P_i , $i \in J_N$, $J_N = \{1, 2, \dots, N\}$. Задана область розміщення $D_0 = \{x \in R^2, x = (x_1, x_2) | 0 \leq x_j \leq b_{0j}, j \in \{1, 2\}\}$, $b_0 = (b_{01}, b_{02})$ — вектор метричних характеристик прямокутника D_0 .

$D_0 = \bigcup_{i=1}^k D_i$ відповідає об'єднанню всіх k аркушів, на яких можуть бути розміщені елементи, в один прямокутник:

$$D_i = \left\{ x \in R^2, x = (x_1, x_2) | 0 \leq x_1 \leq b_{01}, \sum_{j=1}^{i-1} b_{j2} \leq x_2 \leq \sum_{j=1}^i b_{j2} \right\}.$$

Необхідно розмістити заданий набір елементів P_i в області D_0 з можливістю поворотів прямокутників на кут $\frac{\pi}{2}$ таким чином, щоб заповнення областей D_i в напрямку Ox_1 було мінімальним або коефіцієнт заповнення областей D_i був максимальним. Математичні моделі на основі Φ -функцій мають вигляд:

$$X^* = \arg \min_{X \in W \subset R^{2N+1}} F_1(X), F_1(X) = d, d = b_{01}, \quad (1)$$

тут b_{01} являє собою змінну величину, або

$$X^* = \arg \max_{X \in W \subset R^{2N+1}} F_2(X), F_2(X) = \frac{\sum_{j=1}^m a_{j1} \cdot a_{j2}}{b_{01} \cdot b_{02}}, \quad (2)$$

де $X = (u, v, d)$, $u = (u_1, u_2, \dots, u_l)$, $v = (v_1, v_2, \dots, v_l)$ — параметри розміщення, які визначають полюси та ортогональну орієнтацію елементів відповідно; m — кількість елементів, які були розміщені; W — область допустимих рішень, яка описується системою нерівностей:

$$\begin{cases} \Phi_{ij}(u_i, u_j, v_i, v_j) \geq 0, & i, j \in J_m, i < j, \\ \Phi_{0j}(u_0, u_j, v_j) \geq 0, & j \in J_m. \end{cases}$$

Тут Φ_{ij} — Φ -функція [3–5] двох прямокутників P_i та P_j із урахуванням їх ортогональної орієнтації, Φ_{0j} — Φ -функція об'єктів $D_0^* = cl(R^2 \setminus D_0)$ та P_j .

Область допустимих рішень задачі можна описати за допомогою комбінаторної множини композиції перестановок PW_m [6]. Кількість всіх можливих перестановок прямокутників P_i дорівнює $N!$. Різноманітні повороти прямокутників на кут $\frac{\pi}{2}$ описується за допомогою впорядкування лінійних розмірів P_i . Перестановка $a_{ii} = (a_{ik_{i1}}, a_{ik_{i2}})$, де $\{k_{i1}, k_{i2}\}$ — перестановка елементів індексної мно-

жини J_n , $t \in J_M$, $M = 2!$, відповідає всім можливим поворотам прямокутника на кут $\frac{\pi}{2}$, $i \in J_N$.

Застосування комбінаторної множини PW_m дає можливість використання її специфічних комбінаторних властивостей для методу околів, що звужуються [7], при розв'язанні поставленої задачі.

3. Метод розв'язання задачі

Для розв'язання задачі використано метод, який заснований на комбінації послідовно-одиначного алгоритму та методу околів, що звужуються (МОЗ). Приведемо основні кроки алгоритму.

1. Генеруються послідовності прямокутників, які треба розмістити, та послідовності їх лінійних розмірів.

2. У відповідності до отриманих послідовностей прямокутники розміщуються у заданій області за допомогою алгоритму на основі методу послідовно-одиначного розміщення. На кожному кроці цього алгоритму будується множина «контейнерів» C_i — вільних місць прямокутної форми, в які може бути розміщено поточний прямокутник. Після розміщення ті контейнери, які перетинаються з розміщеним прямокутником, або повністю включаються в інший контейнер, або розбиваються на частини, або видаляються. Наприкінці кожного кроку контейнери сортуються в залежності від вказаного критерію якості. Алгоритм закінчується у разі, якщо всі прямокутники розміщені або немає контейнерів, у які можна розмістити решту прямокутників. В результаті отримуємо локальний екстремум задачі. Таким чином, кожній згенерованій послідовності прямокутників та їх лінійних розмірів ставиться у відповідність локальний екстремум задачі.

3. Обчислюються та зберігаються значення цільової функції в отриманих локальних екстремумах.

4. Генеруються нові послідовності прямокутників та обчислюються значення цільової функцій (аналогічно крокам 1-3). За допомогою методу околів, що звужуються, серед знайдених послідовностей обираються найбільш «перспективні».

5. Генеруються нові послідовності прямокутників в околицях визначених «перспективних» послідовностей.

6. У разі відсутності покращення значення цільової функції відбувається звуження околів пошуку оптимального рішення.

7. Якщо радіус околів досягає мінімального значення, то пошук припиняється.

8. Серед знайдених рішень задачі обирається таке, що найкраще задовольняє заданий критерій оптимальності.

4. Обчислювальні експерименти

Було розроблено програмний засіб, який реалізує запропонований алгоритм. За допомогою цього засобу проводились обчислювальні експерименти та склалися графічні відображення отриманих результатів.

Обчислювальні експерименти проводились у три етапи. На першому етапі виконувалось розміщення однієї заданої послідовності прямокутників. На другому етапі випадковим чином генерувались послідовності прямокутників. Кожна згенерована випадкова послідовність розміщувалась, обчислювалось значення цільової функції. Поміж всіх результатів обирався найкращий. На третьому етапі для розв’язання задачі використовувався метод околів, що звужуються. Результати для функції цілі вигляду (2) приведено нижче в табл. 1:

Таблиця 1

Кількість прям-ків	Коефіцієнт заповнення		
	1 посл.	100 посл.	МОЗ
А) 15	0,84	0,89	0,94
Б) 25	0,82	0,88	0,92
В) 45	0,76	0,97	0,99
Г) 50	0,85	0,89	0,91
Д) 59	0,84	0,91	0,93

Графічне відображення найкращих результатів приведено на рис. 3.

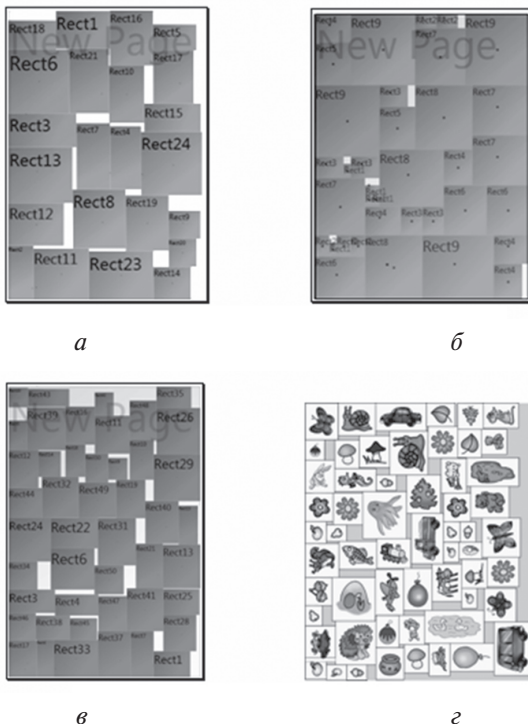


Рис. 3. Графічне відображення найкращих результатів для задач з функцією цілі (2)

Вхідні дані — лінійні розміри прямокутників — кожної задачі генерувались випадковим чином. Площа області розміщення вибиралась такою, яка б могла вмістити всі прямокутники.

Результати розв’язання задачі для функції цілі вигляду (1) приведено нижче в табл. 2.

Таблиця 2

Кількість прям-ків	b_{02}	X^*		
		1 посл.	100 посл.	МОЗ
А) 15	953	867	704	680
Б) 25	1253	1008	953	892
В) 45	400	500	450	450
Г) 50	1822	1471	1383	1348

Графічне відображення найкращих результатів приведено на рис. 4.

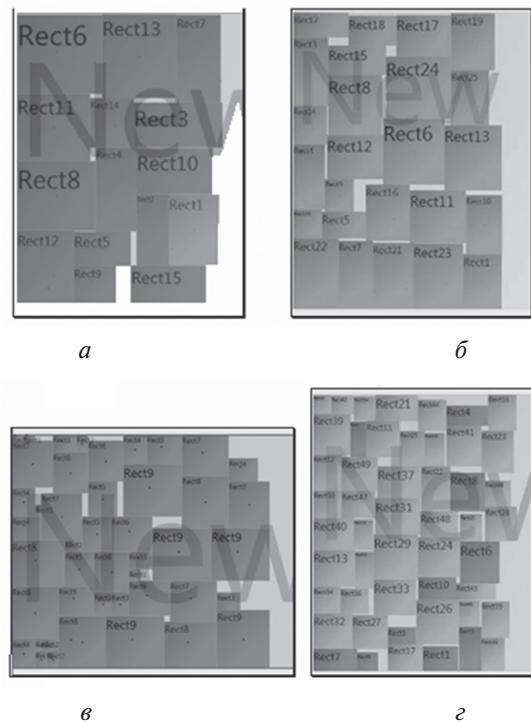


Рис. 4. Графічне відображення найкращих результатів для задач з функцією цілі (1)

Використання запропонованого методу дозволяє задавати декілька аркушів, на які треба розмістити прямокутники з P_i . Для цього в початковий перелік контейнерів C_i додаються аркуші заданого розміру. Аркуші розміщуються один під одним. Далі розв’язання задачі продовжується згідно з алгоритмом. Результати розв’язання задачі з декількома аркушами для функції цілі вигляду (2) приведено нижче в табл. 3.

Таблиця 3

N	N_A	Коефіцієнт заповнення			$t(c)$	N^*
		1 посл.	100 посл.	МОЗ		
А) 50	2	0,83	0,9	0,94	0,01	26
					0,46	25
					894	27
Б) 90	3	0,86	0,91	0,93	0,015	49
					0,9	46
					5620	49

В таблиці N_A — кількість аркушів, t — час розв’язання задачі, N^* — кількість розміщених

прямокутників відповідно для однієї послідовності, 100 послідовностей та при використанні метода околів, що звужуються.

Графічне відображення найкращих результатів приведено на рис. 5.

Найкращий результат було досягнуто при використанні метода околів, що звужуються, але процес розв'язання вимагав найбільше часу. Розміщення 100 послідовностей і вибір з них найкращих теж дало непоганий результат. При цьому час розв'язання був прийнятний — близько 2 секунд для 100 прямокутників. Поліпшити результат можна шляхом збільшення числа послідовностей, які треба розміщати, але це призведе також і до збільшення часу розв'язання. Найгірший результат досягнуто при розміщенні однієї випадкової послідовності прямокутників. Однак одну послідовність можна вибирати, не випадково відсортувавши прямокутники за зростанням площі, довжини діагоналі, тощо. Така модифікація також може поліпшити результат розміщення.

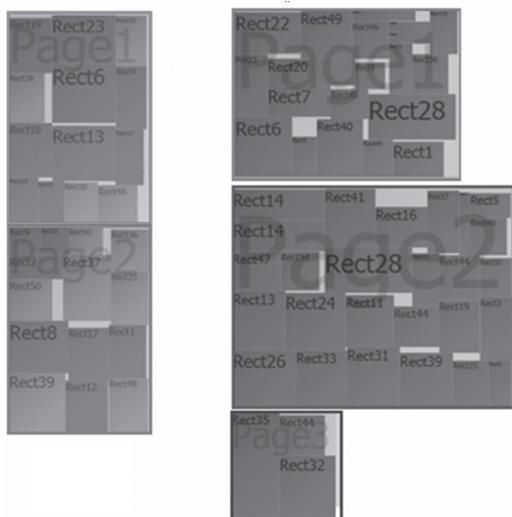


Рис. 5. Графічне відображення найкращих результатів для задач з функцією цілі (1) при використанні кількох аркушів

Висновки

Побудована в роботі математична модель, метод розв'язання задачі розміщення прямокутних елементів та його програмна реалізація можуть бути використані у якості оптимізаційного ядра у складі математичного забезпечення інтелектуальних систем автоматизації видавничої роботи, зокрема, при компонованні блоків реклами в газетній продукції, розміщенні ілюстрацій, розміщенні наліпок при виробництві різноманітних дитячих книжок, тощо.

Використання програмного засобу, створеного на базі запропонованих моделей, методів та алгоритмів, дозволить мінімізувати залишки матеріалу, скоротити час підготовки поліграфічного матеріалу до друку та більш дорогий труд людини. Крім того, отримане рішення можна використовувати як початкове і надавати користувачу можливість в діалоговому режимі змінити деякі параметри розміщення.

Окреслені недоліки запропонованого алгоритму, визначені шляхи подальших досліджень, серед яких: урахування більшого числа особливостей задачі розміщення графічного матеріалу, покращення швидкості роботи методу, покращення інтерфейсу програмного засобу, інтеграція із відомими поліграфічними пакетами та інше. Крім того ведуться дослідження щодо покращення послідовно-одиначного алгоритму таким чином, щоб мінімізувати кількість та загальну площу так званих «дірок» між розміщеними елементами в середині області розміщення.

Список літератури: 1. Дурняк, Б. В. Системний аналіз та оптимізація параметрів книжкових видань: Монографія [Текст] / Б. В. Дурняк, І. В. Піх, В. М. Сеньківський. — Львів: Українська академія друкарства, 2006. — 197 с. 2. Пикок, Джон. Издательское дело: Пер. с англ. 2-е издание, исправленное и дополненное [Текст] / Д. Пикок. — М.: Издательство ЭКОМ, 2002. — 424 с.: илл. 3. Стоян, Ю. Г. Об одном обобщении функции плотного размещения [Текст] / Ю. Г. Стоян // Доклады АН УССР. — 1980. — N 8. — С. 70–74. 4. Стоян, Ю. Г. Ф-функция n -мерных параллелепипедов [Текст] / Ю. Г. Стоян, Н. И. Гиль, М. С. Муравьева // Доповіді НАН України. — 2005. — N 3. — С. 22–27. 5. Размещение n -мерных параллелепипедов с возможностью изменения их ортогональной ориентации в n -мерном параллелепипеде // Материалы 13-го международного молодежного форума «Радиоэлектроника и молодежь в 21 веке», 30 марта 2009 г. — Х.: ХНУРЕ, 2009. — 314 с. 6. Гребенник, И. В. Решение некоторых экстремальных задач на множестве композиций перестановок [Текст] / И. В. Гребенник, А. В. Баранов // Радиотехника. — 2007. — Вып. 149. — С. 12–17. 7. Стоян, Ю. Г. Решение некоторых многоэкстремальных задач методом сужающихся окрестностей [Текст] / Ю. Г. Стоян, В. З. Соколовский. — К.: Наук. думка, 1980. — 208 с.

Надійшла до редколегії 23.02.2010 р.

УДК 519.85

Размещение прямоугольных графических элементов в полиграфическом производстве / А. В. Баранов, Д. В. Грицай, И. В. Гребенник, А. В. Панкратов, А. М. Чугай // Бионика интеллекта: научн.-техн. журнал. — 2010. — № 1 (72). — С. 29–32.

Рассматривается задача размещения набора графических элементов (иллюстраций, блоков рекламы, наклеек) прямоугольной формы с возможностью изменения их ортогональной ориентации на листах материала при раскрое материала для издательско-полиграфического производства. Построена математическая модель, предложен метод на основе комбинации последовательно-одиначного алгоритма и метода сужающихся окрестностей. Проведены вычислительные эксперименты.

Ил. 5. Библиогр.: 7 назв.

UDK 519.85

Placement of the rectangular graphical elements in printing production / A. V. Baranov, D. V. Gritsay, I. V. Grebennik, A. V. Pankratov, A. M. Chugai / Bionics of Intelligence: Sci. Mag. — 2010. — № 1 (72). — P. 29–32.

The given article is devoted to the problem of packing a set of graphical elements (images, blocks of ads, labels) of rectangular shape with the ability to change their orthogonal orientation to the sheet of material when cutting material for publishing in printing production. Mathematical model and solution method based on a combination of series-single algorithm and the method of shrinking neighborhoods are proposed. Computational experiments are implemented.

Fig. 5. Ref.: 7 items.

УДК 004.032 26



НЕЙРОННАЯ СЕТЬ И АЛГОРИТМ ЕЕ ОБУЧЕНИЯ ДЛЯ АНАЛИЗА НЕЗАВИСИМЫХ КОМПОНЕНТ В ЗАДАЧАХ ОБРАБОТКИ ИЗОБРАЖЕНИЙ

А.Г. Ведмедь¹, В.П. Машталир², Е.С. Сакало³

^{1,2,3}ХНУРЭ, г. Харьков, Украина, mashtalir@kture.kharkov.ua

Для решения задачи восстановления изображений предложена нейронная сеть для реконструкции входных сигналов в предположении их взаимной статической независимости. Введенные алгоритмы вычислительно просты и предназначены для обработки информации в последовательном режиме по мере ее поступления.

НЕЙРОННАЯ СЕТЬ, АНАЛИЗ НЕЗАВИСИМЫХ КОМПОНЕНТ, ОБРАБОТКА ИЗОБРАЖЕНИЙ

Введение

Анализ независимых компонент [1] в общем случае связан с проблемой разделения смесей сигналов с априори неизвестными характеристиками, поступающих из различных источников, и в настоящее время достаточно широко используется в задачах обработки биомедицинской информации, данных, получаемых от линейных полей, диагностирования и тестирования сложных электронных систем и, что особенно интересно, при обработке (прежде всего восстановления) изображений, искаженных возмущениями неизвестной природы.

Особенно эффективно анализ независимых компонент может быть реализован на основе нейросетевого подхода [2-7], что создает предпосылки для обработки поступающих данных в реальном времени.

В анализе независимых компонент априори предполагается, что исходный n -мерный сигнал $x(t) = (x_1(t), x_2(t), \dots, x_n(t))^T$ (здесь t – непрерывное время) воспринимается датчиками не в «чистом» виде, а в форме линейной комбинации так, что

$$y_i(t) = \sum_{j=1}^n a_{ij} x_j(t), i = 1, 2, \dots, n$$

или в векторной форме

$$y(t) = Ax(t),$$

где $A \in R^{n \times n}$; $\det A \neq 0$; $y(t) = (y_1(t); y_2(t), \dots; y_n(t))^T$. При этом полагается, что как сигналы $x_j(t)$, так и матрица A неизвестны. Задача состоит в разделении (реконструкции) входных сигналов в предположении их взаимной статической независимости и существования матрицы A^{-1} , обратной к неизвестной смешивающей матрице A . Что касается обработки изображений, то достаточно часто возникает задача не только их «очистки» – фильтрации от различного рода искажений и изменений, но и разделения смеси различных изображений, когда разные образы накладываются один на другой.

Именно в такой форме проблема впервые рассмотрена и решена Дж. Херольтом и К. Джуттенном

[8-10], хотя введенная ими рекуррентная разделяющая нейронная сеть на сегодня считается недостаточно эффективной.

Рассматривая данную задачу с позиций нейросетевого подхода, можно говорить о синтезе архитектур и модернизированных алгоритмов обучения, обеспечивающих преобразование наблюдаемого векторного сигнала $y(t) = (y_1(t), y_2(t), \dots, y_n(t))^T$ в некий новый вектор $s(t) = (s_1(t), s_2(t), \dots, s_n(t))^T$ с независимыми компонентами $s_i(t)$, при этом статистически независимая от других компонент $s_i(t)$ должна быть связана в некотором смысле с соответствующей компонентой $x(t)$.

На рис. 1 приведена обобщенная схема нейросетевой реализации анализа независимых компонент.

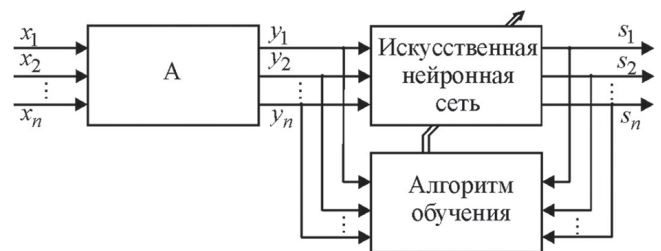


Рис. 1. Нейросетевая реализация анализа независимых компонент

Элементарные нейронные структуры, рассмотренные выше, могут быть обобщены на случай многомерного сигнала произвольной размерности n , при этом основные соотношения, описывающие данную ситуацию, могут быть записаны в виде

$$s(t) = Wy(t),$$

$$y(t) = Ax(t),$$

где $W = \{w_{ij}\} \in R^{n \times n}$ – матрица настраиваемых синтаксических весов и приведена на рис. 2.

Как видно, данная архитектура полностью совпадает с конструкцией самоорганизующейся карты Т. Кохонена с одинаковым числом входов и выходов.

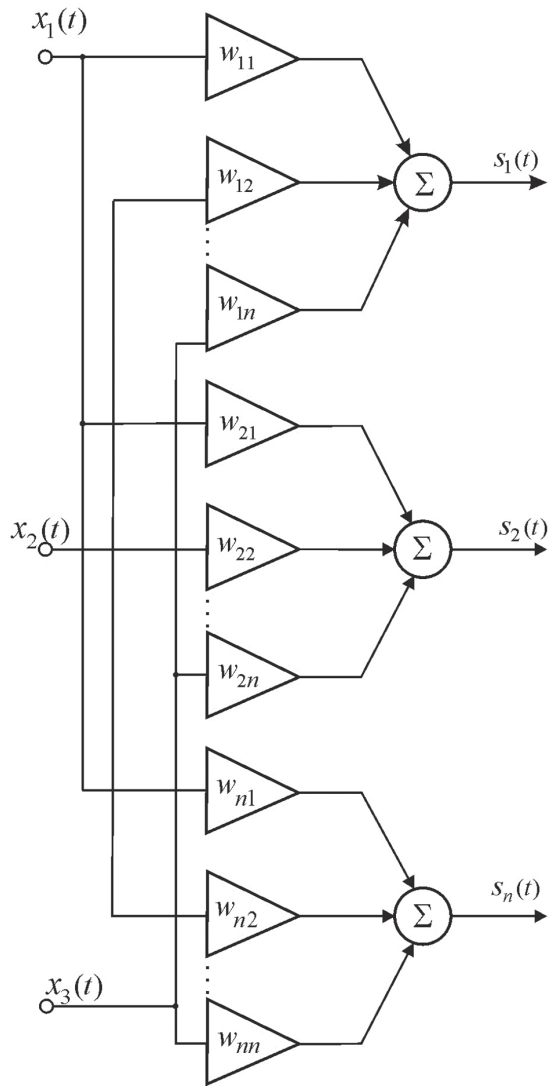


Рис. 2. Нейронная сеть для анализа независимых компонент

1. Элементарная нейронная сеть для анализа независимых компонент и алгоритмы ее обучения

В качестве базового элемента нейросетевой системы для анализа независимых компонент рассмотрим элементарную двухвходовую подсистему, предназначенную для разделения ненаблюдаемых сигналов $x_1(t)$ и $x_2(t)$. В настоящее время для решения этой задачи используются две основные архитектуры [2,7], приведенные на рис. 3 (а, б).

Пунктиром обведена часть, недоступная наблюдению, при этом неизвестные веса смешивающей матрицы $a_{11}, a_{12}, a_{21}, a_{22}$ должны быть оценены с помощью настраиваемых синаптических весов w_{12} и w_{21} путем соответствующей обработки сигналов $y_1(t), y_2(t), s_1(t), s_2(t)$.

Обучение структур, приведенных на рис. 3, сводится к нахождению в процессе настройки значений синаптических весов w_{12} и w_{21} , а в его основе лежит хеббовское правило [11] в непрерывном времени, модифицированное Джуттенем-Херольдом [9] с учетом статической независимости сигналов $s_1(t)$ и $s_2(t)$.

Поскольку задачи обработки изображений решаются в цифровом виде, следует ввести дискретные алгоритмы обучения, получаемые с помощью цифрового квантования [12]:

$$\begin{cases} w_{12}(k+1) = w_{12}(k) + \eta(k)\varphi(s_1(k))\psi(s_2(k)), \\ w_{21}(k+1) = w_{21}(k) + \eta(k)\varphi(s_2(k))\psi(s_1(k)), \end{cases}$$

где $k = 1, 2, \dots, N$ – текущее дискретное время; $\eta(k)$ – параметр шага алгоритма обучения, выбираемого по аналогии с параметром шага в градиентных алгоритмах оптимизации.

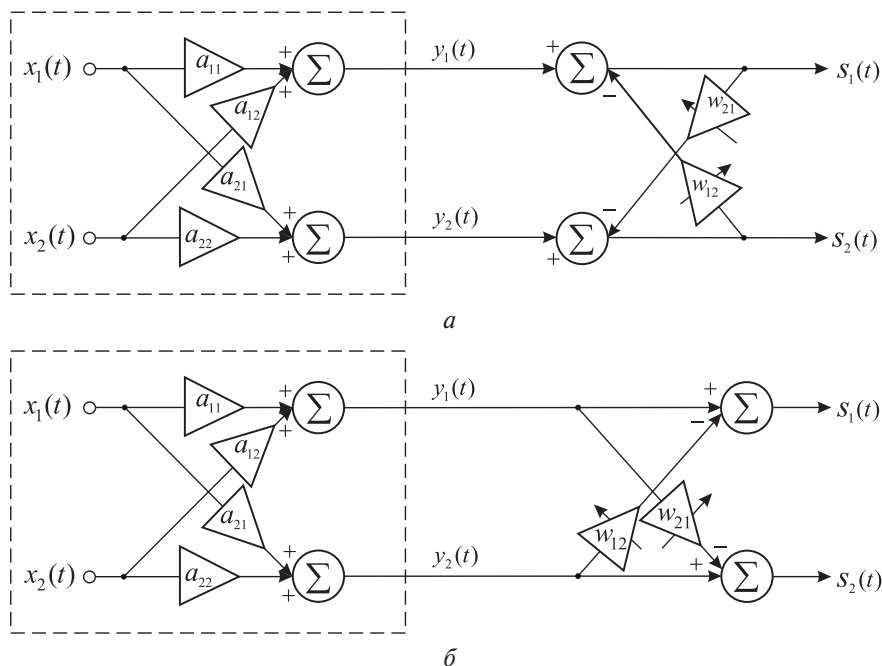


Рис. 3. Элементарная нейронная сеть для анализа независимых компонент

2. Искусственная нейронная сеть для анализа главных компонент и ее обучение

К настоящему времени известно достаточно большое число правил обучения такой сети в непрерывном времени [2, 5, 6, 10, 11], среди которых можно отметить:

– модифицированное правило Чихотского

$$\frac{dW}{dt} = \eta(t)W(I - \varphi(s(t))\psi^T(s(t)))W,$$

где $W(0) = I$; $\varphi(s(t)) = (\varphi(s_1(t)); \varphi(s_2(t)); \dots; \varphi(s_n(t)))^T$, $\psi^T(s(t)) = (\psi(s_1(t)); \psi(s_2(t)); \dots; \psi(s_n(t)))$; $I - (n \times n)$ единичная матрица;

– правило Чихотского-Амари-Янга

$$\frac{dW}{dt} = \eta(t)(I - \varphi(s(t))\psi^T(s(t)))W^{-1},$$

– правило, основанное на естественном градиенте Амари

$$\frac{dW}{dt} = \eta(t)(I - \varphi(s(t))\psi^T(s(t)))W,$$

– правило Кардоссо

$$\frac{dW}{dt} = \eta(t)(I - s(t)s^T(t) - \alpha\varphi(s(t))\psi^T(s(t)) + \beta\psi(s(t))\varphi^T(s(t)))W,$$

$$\frac{dW}{dt} = \eta(t)(I - \varphi(s(t))s^T(t))W,$$

где α и β – некоторые параметры, заданные на интервале $[0, 1]$ и выбираемые обычно из эмпирических соображений.

Введем в рассмотрение алгоритмы дискретного времени, соответствующие указанным выше непрерывным правилам. При этом модифицированному правилу Чихотского соответствует алгоритм

$$W(k+1) = W(k) + \eta(k)W(k)(I - \varphi(s(k))\psi^T(s(k)))W(k),$$

правилу Чихотского-Амари-Янга –

$$W(k+1) = W(k) + \eta(k)(I - \varphi(s(k))\psi^T(s(k)))W^{-1}(k),$$

правилу на базе градиента Амари –

$$W(k+1) = W(k) + \eta(k)(I - \varphi(s(k))\psi^T(s(k)))W(k),$$

правилу Кардоссо –

$$W(k+1) = W(k) + \eta(k)(I - s(k)s^T(k) - \alpha\varphi(s(k)) \times \psi^T(s(k)) + \beta\psi(s(k))\varphi^T(s(k)))W(k)$$

и, наконец,

$$W(k+1) = W(k) + \eta(k)(I - \varphi(s(k))s^T(k))W(k).$$

В классе непрерывных нейронных сетей для анализа независимых компонент весьма эффек-

тивной показала себя сеть Чихотского-Унбеауэна [2], содержащая кроме фильтров высоких частот (ФВЧ) для исключения постоянной компоненты дополнительные фильтры низких частот (ФНЧ) для подавления нежелательных колебаний в выходном сигнале в процессе обучения. Данная сеть описывается правилами обучения вида

$$\frac{d\tilde{w}_{ij}(t)}{dt} = \eta(\beta(t)\varphi(s_i(t))\psi(s_j(t)*h(t)) - \alpha\tilde{w}_{ij}), \quad (1)$$

$$\frac{dw_{ij}(t)}{dt} = \mu\tilde{w}_{ij}, \quad i, j = 1, 2, \dots, n, \quad (2)$$

где $\eta = \frac{1}{\tau_1}$; $\mu = \frac{1}{\tau}$; $\alpha > 0$; $\beta(t) \geq 0$; $\lim_{t \rightarrow \infty} \beta(t) = 0$.

Для решения задач обработки изображений вместо (1) и (2) целесообразно ввести их дискретные аналоги

$$\tilde{w}_{ij}(k+1) = \tilde{w}_{ij}(k) + \eta(k)(\beta(k)\varphi(s_i(k))\psi(s_j(k)) - \bar{s}_j(k)) - \lambda\tilde{w}_{ij}(k), \quad (3)$$

$$\bar{s}_j(k+1) = \bar{s}_j(k) + \eta_s(k)(s_j(k+1) - s_j(k)), \quad (4)$$

$$w_{ij}(k+1) = (1 - \mu)w_{ij}(k) + \mu\tilde{w}_{ij}(k+1), \quad (5)$$

отличающиеся тем, что вместо операции высокочастотной фильтрации в (1) используется операция исключения среднего, вычисляемого с помощью соотношения (4), и реализуемая с помощью блока исключения среднего (БИС), а вместо низкочастотной фильтрации (2) – операция экспоненциального сглаживания (5) с параметром забывания μ , реализуемая с помощью блока экспоненциального сглаживания (БЭС). На рис. 4 приведена схема БИС, а на рис. 5 – БЭС.

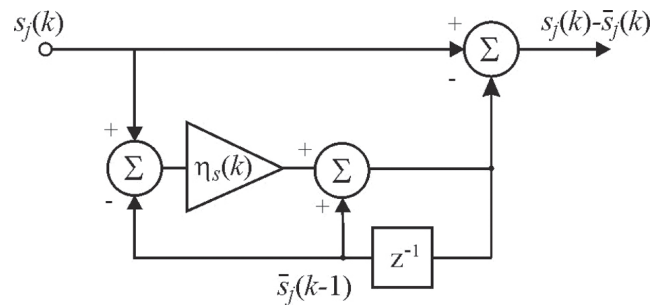


Рис. 4. Блок исключения среднего (БИС)

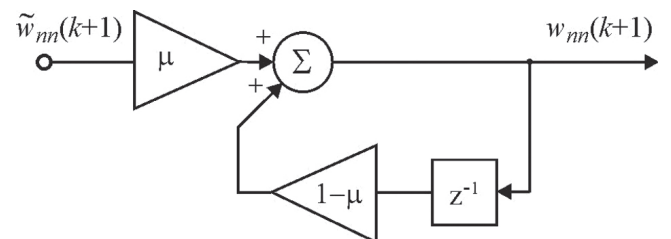


Рис. 5. Блок экспоненциального сглаживания (БЭС)

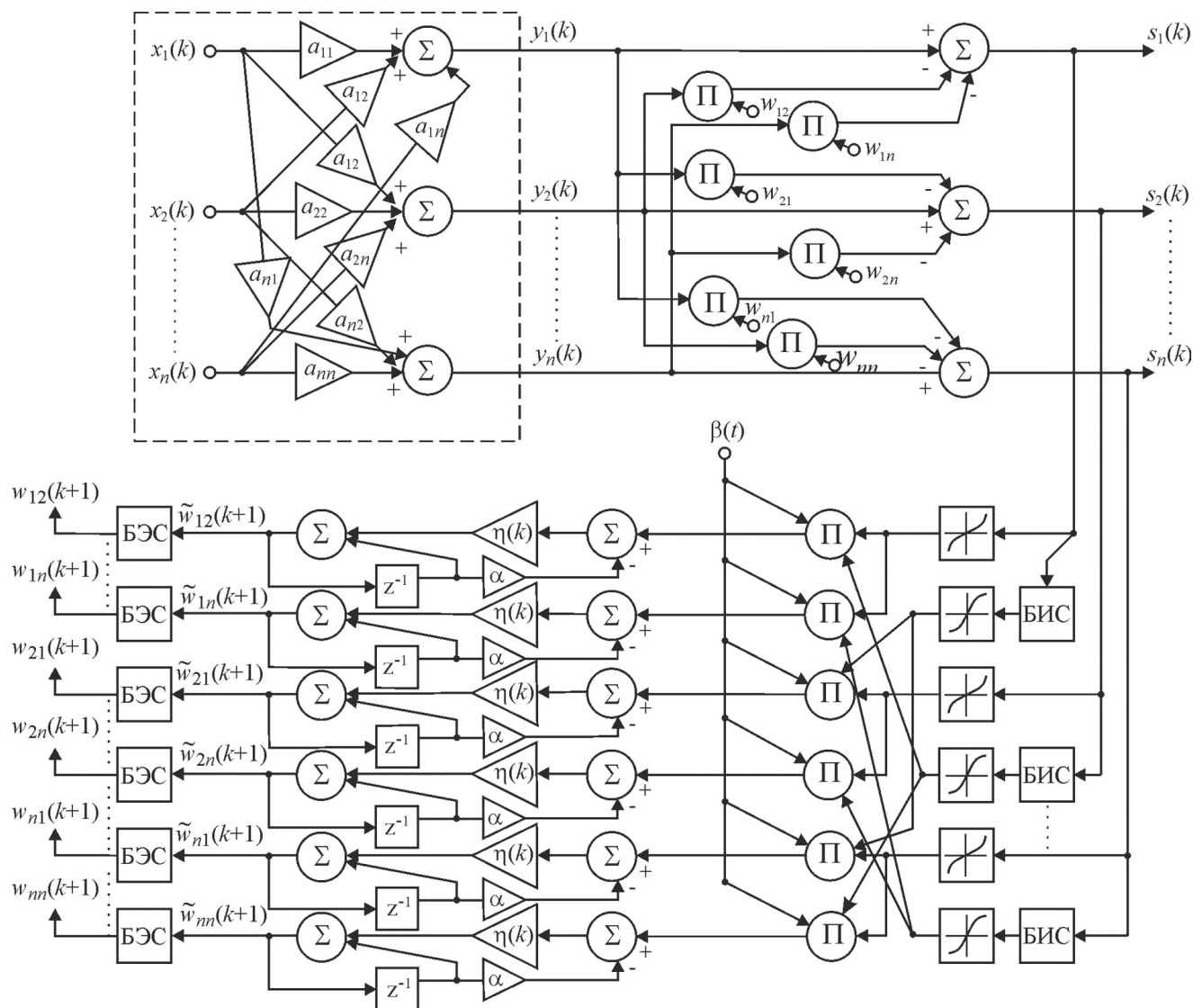


Рис. 6. Нейронная сеть для анализа независимых компонент в дискретном времени

На рис. 6 приведена архитектура нейронной сети для анализа независимых компонент на основе соотношений (3)-(5).

Выводы

Итак, в общем случае задача слепой сепарации может решаться на основе однослойных архитектур, содержащих в качестве нейронов адаптивные линейные ассоциаторы и предназначенных изначально для решения задач анализа главных компонент. Модифицированы алгоритмы обучения, позволяющие в реальном времени настраивать матрицу синаптических весов и обладающие высоким быстродействием и простотой численной реализации. В итоге предложена архитектура сети слепой сепарации и рекуррентная процедура ее обучения, позволяющие обрабатывать сильно “зашумленные” нестационарные сигналы. Предложенная нейронная сеть в процессе своего функционирования реализует лишь три элементарные операции сло-

жения, умножения и сдвига назад, что делает ее реализацию крайне простой. Особенно актуальным это становится в случаях, когда обработка производится в последовательном on-line режиме.

Список литературы: 1. Comon P. Independent component analysis. A new concept? // Signal Processing. – 1994. – Vol. 36. – P. 287-314. 2. Cichocki A., Unbehauen R. Neural networks for optimisation and signal processing. – Stuttgart: Teubner, 1993. – 526p. 3. Karhunen J., Oja E., Wang I., Vigario R., Joutsensablo J. A class of neural network for independent component analysis // IEEE Trans. on Neural Networks. – 1995. – No 8 – P.487-504. 4. Girolami M., Cichocki A. Amari S.I. A common neural network model for exploratory data analysis and independent component analysis // IEEE Trans. on Neural Networks. – 1997. – No 9. – P.1495 – 1501. 5. Cardoso J.-F. Blind signal separation: Statistical principles // Proc. IEEE – 1998. – Vol. 86. – P.2009-2025. 6. Amari S., Cichocki A. Adaptive blind signal processing – neural network approaches // Proc. IEEE – 1998. – Vol. 86. – P.2026-2048. 7. Girolami M. Self-organizing mental networks. Independent component analysis and blind source

separation. — Berlin: Springer Verlag, 1999. — 271p. **8. Jutten C., Herault J.** Independent component analysis (INCA) versus principal component analysis // J.L. Lacoume (Ed.). — “Signal Processing 4”. — Amsterdam: Elsevier, 1988. — P.643-646. **9. Jutten C., Herault J.** Blind separation of sources. Part I: An adaptive algorithm based on neuromimetic architecture // Signal Processing. — 1991. — Vol. 24. — P.1-10. **10. Осовский, С.** Нейронные сети для обработки информации [Текст] / С. Осовский. — Москва: Финансы и статистика, 2002. — 314с. **11. Haykin S.** Neural networks. A comprehensive foundation. — N.J.: Prentice Hall, Inc., 1999. — 842 p. **12. Изерман, Р.** Цифровые системы управления [Текст] / Р. Изерман. — Москва: Мир, 1984. — 541 с. **13. Yip K.-H., Guan L., Liu W.** A recursive soft-decision approach to blind image deconvolution // IEEE Trans. on Signal Processing. — 2003. — Vol. 51. — P. 515-526.

Поступила в редколлегию 24.02.2010 г.

УДК 004.032 26

Нейрона мережа та алгоритм її навчання для аналізу незалежних компонент в задачах обробки зображень / В.П. Машталір, А. Е. Ведмідь, Е.С. Сакало // Біоніка інтелекту: наук.-техн. журнал. — 2010. — № 1 (72). — С. 33–37.

Для вирішення задачі відновлення зображень в роботі запропонована та проаналізована нейронна мережа для реконструкції вхідних сигналів в припущенні їх взаємної статичної незалежності.

Л. 6. Бібліогр.: 13 найм.

UDC 004.032 26

Neural network and learning algorithm for independent component analysis in processing of images / A.G. Vedmed V.P. Mashtalir, E.S. Sakalo // Bionics of Intelligence: Sci. Mag. — 2010. — № 1 (72). — P. 33–37.

For the decision of task of proceeding in images in process offered and analysed a neuron network is for the reconstruction of entrance signals in supposition them mutual static independence.

Fig. 6. Ref.: 13 items.

УДК 004.93



КРИТЕРИИ ИНДИВИДУАЛЬНОЙ ИНФОРМАТИВНОСТИ И МЕТОДЫ ОТБОРА ЭКЗЕМПЛЯРОВ ДЛЯ ПОСТРОЕНИЯ ДИАГНОСТИЧЕСКИХ И РАСПОЗНАЮЩИХ МОДЕЛЕЙ

С.А. Субботин

Запорожский национальный технический университет, г. Запорожье, Украина, subbotin@zntu.edu.ua

С целью автоматизации процесса построения диагностических и распознающих моделей предложен комплекс критериев индивидуальной и групповой информативности экземпляров, методы формирования выборки, позволяющие оценивать качество, а также автоматизировать процесс формирования репрезентативных выборок по отношению к исходной выборке.

РАСПОЗНАВАНИЕ ОБРАЗОВ, ДИАГНОСТИКА, ВЫБОРКА, ЭКЗЕМПЛЯР, МЕТОДЫ ФОРМИРОВАНИЯ ВЫБОРОК

Введение

В задачах неразрушающей диагностики и распознавания образов по признакам одной из главных проблем является сокращение размерности выборки данных для построения модели принятия решений на основе машинного обучения по прецедентам [1].

Наряду с использованием методов отбора информативных признаков [1, 2], ставших традиционным инструментом сокращения размерности, размерность данных также возможно сократить путем выделения наиболее значимых для решения задачи распознавания экземпляров.

Известные методы извлечения выборок [1, 3–5], как правило, не обеспечивают гарантии выделения наиболее важных для задачи экземпляров, а также приводят к формированию избыточных выборок, поскольку основаны на случайном покрытии множества возможных решений. Поэтому представляется актуальной разработка математических критериев и методов, позволяющих оценивать значимость экземпляров и управлять процессом их выбора.

Целью данной работы является создание комплекса критериев оценивания индивидуальной и групповой информативности экземпляров, а также методов их выбора, позволяющих автоматизировать процесс формирования выборок для построения диагностических и распознающих моделей.

1. Постановка задачи

Пусть задана исходная выборка $\langle X, Y \rangle$. Необходимо извлечь из нее обучающую выборку (подвыборку) $\langle x, y \rangle$, которая является репрезентативной по отношению к исходной выборке, то есть отображает все наиболее важные ее свойства.

Обозначим: x^s – s -ый экземпляр выборки; x_j^s – значение j -го признака s -го экземпляра; y^s – значение выходного признака; сопоставленное s -му экземпляру выборки; S – число экземпляров выборки; S^q – количество экземпляров выборки, принадлежащих к q -му классу; N – число входных признаков; K – число классов.

2. Критерии индивидуальной информативности экземпляров

Задача критериев индивидуальной информативности экземпляров заключается в том, чтобы для каждого экземпляра можно было оценить его ценность для процесса построения диагностической или распознающей модели. Очевидно, что наибольшую ценность будут иметь те экземпляры, которые расположены на внутренних и внешних границах классов, а также близко расположены к центрам классов и кластеров. Соответственно зададим критерии, отражающие данные соображения.

Критерий индивидуальной информативности s -го экземпляра относительно внутренней границы классов определим как:

$$I_{\hat{G}}^s = \frac{1}{S-1} \sum_{p=1}^S \left\{ e^{-\alpha_{y^s, y^p} \sum_{j=1}^N (x_j^s - x_j^p)^2} \mid s \neq p, y^s \neq y^p \right\}$$

$$\text{или } I_{\hat{G}'}^s = \frac{1}{S-1} \sum_{p=1}^S \left\{ e^{-\alpha_{y^s, y^p} \min_{j=1,2,\dots,N} \{x_j^s - x_j^p\}} \mid s \neq p, y^s \neq y^p \right\}$$

$$\text{или } I_{\hat{G}}^s = \max_{p=1,2,\dots,S} \left\{ e^{-\alpha_{y^s, y^p} \sum_{j=1}^N (x_j^s - x_j^p)^2} \mid s \neq p, y^s \neq y^p \right\}$$

$$\text{или } I_{\hat{G}'}^s = \max_{p=1,2,\dots,S} \left\{ e^{-\alpha_{y^s, y^p} \min_{j=1,2,\dots,N} \{x_j^s - x_j^p\}} \mid s \neq p, y^s \neq y^p \right\},$$

где

$$\alpha_{k,q} = \frac{1}{S^k S^q \sum_{s=1}^S \sum_{p=1}^S \left\{ \sum_{j=1}^N (x_j^s - x_j^p)^2 \mid (y^s = k, y^p = q) \vee (y^s = q, y^p = k) \right\}}.$$

Данный критерий будет принимать значения в диапазоне от нуля до единицы: его значение будет тем больше, чем ближе экземпляр расположен к границе между разными классами.

Критерий индивидуальной информативности s -го экземпляра относительно его удаленности определим:

– относительно границы класса как:

$$I_U^s = 1 - \min_{p=1,2,\dots,S} \left\{ e^{-\alpha \sum_{j=1}^N (x_j^s - x_j^p)^2} \mid s \neq p, y^s = y^p \right\}$$

$$\text{или } I_{U'}^s = 1 - \min_{p=1,2,\dots,S} \left\{ e^{-\alpha \sum_{j=1,2,\dots,N} \min \{ |x_j^s - x_j^p| \}} \mid s \neq p, y^s = y^p \right\}.$$

$$\text{где } \alpha_k = \frac{1}{\max_{\substack{s=1,2,\dots,S; \\ p=s+1,\dots,S}} \left\{ \sum_{j=1}^N (x_j^s - x_j^p)^2 \mid y^s = y^p \right\}}.$$

– относительно внешней границы выборки как:

$$I_{U''}^s = 1 - \min_{p=1,2,\dots,S} \left\{ e^{-\alpha \sum_{j=1}^N (x_j^s - x_j^p)^2} \mid s \neq p \right\}$$

$$\text{или } I_{U'''}^s = 1 - \min_{p=1,2,\dots,S} \left\{ e^{-\alpha \sum_{j=1,2,\dots,N} \min \{ |x_j^s - x_j^p| \}} \mid s \neq p \right\}.$$

$$\text{где } \alpha = \frac{1}{\max_{\substack{s=1,2,\dots,S; \\ p=s+1,\dots,S}} \left\{ \sum_{j=1}^N (x_j^s - x_j^p)^2 \right\}}.$$

Данный критерий будет принимать значения в диапазоне от нуля до единицы: его значение будет тем больше, чем дальше экземпляр расположен по отношению к остальным экземплярам класса или выборки, то есть критерий будет выявлять редкие нетипичные случаи, а также экземпляры, расположенные на внешних границах классов.

Критерий индивидуальной информативности s -го экземпляра относительно его близости к центру класса (кластера) определим как:

$$I_O^s = \frac{1}{S^{y^s} - 1} \sum_{p=1}^S \left\{ e^{-\sum_{j=1}^N (x_j^s - x_j^p)^2} \mid s \neq p, y^s = y^p \right\}$$

$$\text{или } I_{O'}^s = \frac{1}{S^{y^s} - 1} \sum_{p=1}^S \left\{ e^{-\sum_{j=1,2,\dots,N} \min \{ |x_j^s - x_j^p| \}} \mid s \neq p, y^s = y^p \right\}.$$

Данный критерий будет принимать значения в диапазоне от нуля до единицы: его значение будет тем больше, чем ближе экземпляр расположен к “остову” класса.

Интегральный показатель информативности s -го экземпляра определим как:

$$\hat{I}^s = \max(I_{\bar{G}}^s, I_{\hat{G}}^s, I_O^s, I_U^s)$$

$$\text{или } \hat{I}'^s = \max(I_{\bar{G}'}^s, I_{\hat{G}'}^s, I_{O'}^s, I_{U'}^s)$$

$$\text{или } I^s = \max(I_{\bar{G}}^s, I_{\hat{G}}^s, I_O^s, I_U^s, I_{\bar{G}'}^s, I_{\hat{G}'}^s, I_{O'}^s, I_{U'}^s).$$

Данный критерий будет принимать значения в диапазоне от нуля до единицы: чем больше будет его значение, тем значимее s -ый экземпляр для построения модели, поскольку он либо находится

на границе между классами, либо является уникальным наблюдением, либо находится на внешней границе класса, либо соответствует “остову” класса (близок к центру кластера).

2. Критерии групповой информативности экземпляров

Задача критериев групповой информативности экземпляров заключается в том, чтобы, с одной стороны, охарактеризовать качество выборки, а с другой стороны, определить ее важнейшие характеристики для сравнения с другими выборками для решения той же задачи.

Критерий групповой информативности экземпляров выборки относительно описания внутренней границы классов определим как:

$$I_{\bar{G}} = \frac{1}{S} \sum_{s=1}^S I_{\bar{G}}^s \quad \text{или} \quad I_{\bar{G}'} = \frac{1}{S} \sum_{s=1}^S I_{\bar{G}'}^s,$$

$$\text{или } I_{\hat{G}} = \frac{1}{S} \sum_{s=1}^S I_{\hat{G}}^s \quad \text{или} \quad I_{\hat{G}'} = \frac{1}{S} \sum_{s=1}^S I_{\hat{G}'}^s.$$

Данный критерий будет принимать значения в диапазоне от нуля до единицы: его значение будет тем больше, чем больше выборка содержит экземпляров, расположенных на границе между разными классами.

Критерий групповой информативности выборки относительно описания внешних границ классов определим как:

$$I_U = \frac{1}{S} \sum_{s=1}^S I_U^s \quad \text{или} \quad I_{U'} = \frac{1}{S} \sum_{s=1}^S I_{U'}^s.$$

Данный критерий будет принимать значения в диапазоне от нуля до единицы: его значение будет тем больше, чем больше выборка содержит экземпляров, расположенных на внешних границах классов.

Критерий групповой информативности экземпляров выборки относительно описания центров классов (кластеров) определим как:

$$I_O = \frac{1}{S} \sum_{s=1}^S I_O^s \quad \text{или} \quad I_{O'} = \frac{1}{S} \sum_{s=1}^S I_{O'}^s.$$

Данный критерий будет принимать значения в диапазоне от нуля до единицы: его значение будет тем больше, чем больше выборка содержит экземпляров, расположенных близко к “остовам” классов.

Интегральный показатель групповой информативности экземпляров выборки определим как:

$$\bar{I} = \left\{ \frac{K}{4S} \left(I_{\bar{G}} + I_{\hat{G}} + I_O + I_U \right) \mid S \geq K, K \geq 2 \right\}.$$

Данный критерий будет принимать значения в диапазоне от нуля до единицы: чем больше будет его значение, тем более ценной является выборка

для построения модели, поскольку она менее избыточна и содержит наиболее важные экземпляры для аппроксимации границ и центров классов.

3. Переборные методы формирования выборок

Для формирования подвыборки из исходной выборки предлагается использовать *модифицированный метод полного перебора* [1], который будет включать следующие этапы:

1. Инициализация задать исходную выборку $\langle X, Y \rangle$. Оценить индивидуальные информативности для каждого экземпляра исходной выборки. Рассчитать значение критерия качества исходной выборки \bar{I} .

2. Генерация: сгенерировать все возможные подвыборки $\langle x(k), y(k) \rangle$ исходной выборки $\langle X, Y \rangle$, где k – номер подвыборки, $x(k), y(k)$ – соответственно экземпляры k -ой выборки и сопоставленные им значения выходного признака.

3. Для каждой сгенерированной подвыборки рассчитать значение критерия качества $\bar{I}(k)$.

4. Среди сформированных подвыборок $\{\langle x(k), y(k) \rangle\}$ в качестве решения $\langle x, y \rangle$ выбрать ту подвыборку $\langle x(p), y(p) \rangle$, которая наилучшим образом соответствует заранее заданному критерию выбора решения.

Предлагается использовать один из следующих критериев выбора решения:

– критерий максимума качества формируемой выборки: $p = \arg \max_k \{ \bar{I}(k) \}$;

– критерий максимального соответствия качества формируемой выборки качеству исходной выборки: $p = \arg \min_k \{ | \bar{I}(k) - \bar{I} | \}$.

Достоинством данного метода является то, что он гарантированно находит лучшее из возможных решений. Недостатком метода является его чрезвычайно высокая вычислительная сложность. Поэтому практическое применение полного перебора возможно только для небольших исходных выборок.

Альтернативой методу полного перебора может служить *модифицированный метод сокращенного перебора* [1] с добавлением и удалением экземпляров. Данный метод может быть представлен как последовательность следующих этапов:

1. Инициализация задать исходную выборку $\langle X, Y \rangle$. Оценить индивидуальные информативности для каждого экземпляра исходной выборки. Рассчитать значение критерия качества исходной выборки \bar{I} . Установить: $k = 1$.

2. Генерация базовой подвыборки $\langle x(1), y(1) \rangle$ исходной выборки $\langle X, Y \rangle$: включить в выборку $\langle x(1), y(1) \rangle$ все экземпляры исходной выборки, для которых:

$$I^s \geq \frac{\alpha}{S} \sum_{s=1}^S I^s,$$

где α – некоторая константа; $0 < \alpha \leq 1$.

3. Добавление экземпляров. Оценить для подвыборки $\langle x(k), y(k) \rangle$ значение показателя качества $\bar{I}(k)$. Если $\bar{I}(k) \leq \bar{I}(k-1)$, то принять: $x(k) = x(k-1)$, $y(k) = y(k-1)$, перейти к этапу 4; в противном случае: если $\bar{I}(k)$ является приемлемым, то перейти к этапу 4, в противном случае: установить: $k = k+1$, $x(k) = x(k-1)$, $y(k) = y(k-1)$; если в исходной выборке имеются экземпляры, отсутствующие в подвыборке $\langle x(k), y(k) \rangle$, то выбрать из них экземпляр с максимальным значением I^s и включить его в подвыборку $\langle x(k), y(k) \rangle$, после чего перейти к этапу 3; если в исходной выборке не осталось экземпляров, отсутствующих в $\langle x(k), y(k) \rangle$, то перейти к этапу 4.

4. Удаление экземпляров. Оценить для подвыборки $\langle x, y \rangle$ значение показателя качества $\bar{I}(k)$. Если $\bar{I}(k)$ является приемлемым и подвыборка $\langle x(k), y(k) \rangle$ – непустая, то установить $k = k+1$, $x(k) = x(k-1)$, $y(k) = y(k-1)$, исключить из $\langle x(k), y(k) \rangle$ экземпляр с минимальным значением I^s , после чего перейти к этапу 4; в противном случае – принять: $x(k) = x(k-1)$, $y(k) = y(k-1)$, перейти к этапу 5.

5. Вернуть в качестве решения выборку $\langle x(k), y(k) \rangle$.

Достоинством данного метода является то, что он сокращает количество рассматриваемых решений за счет включения наиболее перспективных решений в базовую подвыборку, которая затем последовательно дополняется оставшимися экземплярами до тех пор, пока продолжается увеличение групповой информативности выборки, после чего из выборки последовательно удаляются избыточные экземпляры до тех пор, пока групповая информативность оказывается приемлемой.

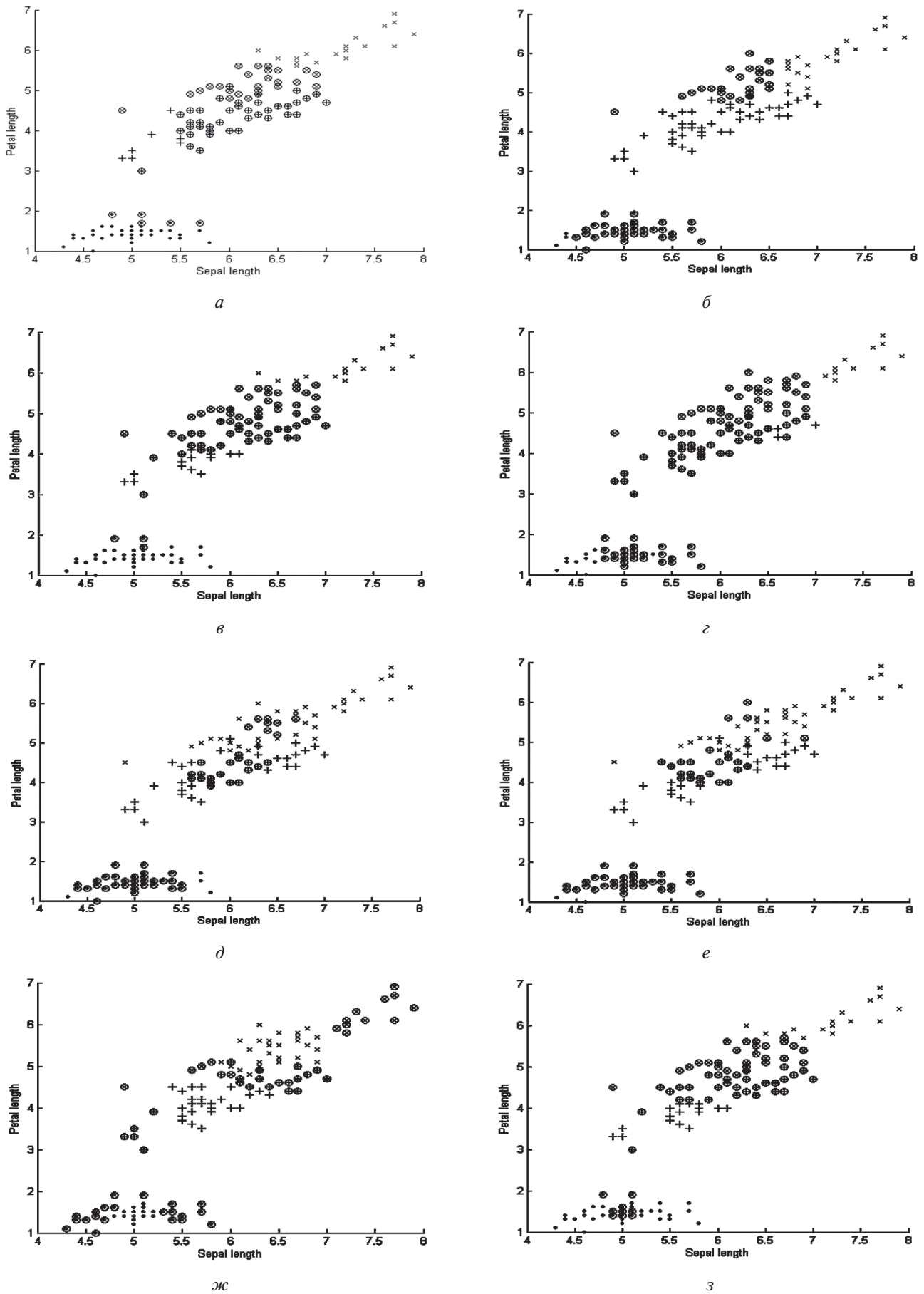
4. Эксперименты и анализ результатов

Для исследования практической применимости предложенных критериев и методов была разработана программа на языке пакета MATLAB, с помощью которой проводились эксперименты. Проведенные эксперименты подтвердили работоспособность предложенного математического обеспечения и его пригодность для решения практических задач диагностики и распознавания образов.

На рис. 1 изображена выборка для известной задачи распознавания ирисов Фишера в пространстве двух признаков (sepal length – длина чашелистика, petal length – длина лепестка).

Здесь значками “.” и “+” и “х”, обозначены, соответственно, экземпляры трех разных классов из исходной выборки, а значком “о” показаны экземпляры, имеющие индивидуальные оценки информативности выше средней по исходной выборке.

Легко видеть, что именно экземпляры, показанные значком “о”, расположены на границах и в центре классов, что наглядно демонстрирует работоспособность предложенных критериев.



ж

з

Рис. 1. Результаты выбора экземпляров на основе критериев:

$$a - I_G^s; \delta - I_G^s; \epsilon - I_G^s; \epsilon - I_G^s; \delta - I_O^s; e - I_O^s; \text{ж} - I_U^s; \text{з} - \hat{I}^s$$

Для сравнения методов формирования выборок оценим их временную сложность.

Для метода полного перебора сложность может быть оценена как $O((2^{S^*}-1)n)$, где S^* – количество экземпляров в исходной выборке, а n – сложность расчета интегрального показателя качества для подвыборки. Эта оценка свидетельствует о практической пригодности полного перебора только для исходных выборок небольшого объема.

Для модифицированного метода сокращенного перебора будем исходить из того, что базовая выборка, формируемая на втором этапе, будет включать не менее половины экземпляров исходной выборки; в свою очередь, на третьем этапе в худшем случае будет сгенерировано $0,5S^*$ подвыборок, а на четвертом этапе – S^* подвыборки. Следовательно, сложность метода можно оценить как $O((2+1,5S^*)n)$. Эта оценка свидетельствует о практической пригодности полного перебора только для исходных выборок большого объема. Таким образом, метод сокращенного перебора по сравнению с методом полного перебора будет работать быстрее в $\frac{2^{S^*}-1}{1,5S^*+2} \approx 2^{S^*-\log_2(1,5S^*)}$ раз.

Выводы

С целью автоматизации процесса построения диагностических и распознающих моделей в работе решена актуальная проблема анализа свойств и формирования обучающих выборок, репрезентативных относительно исходной выборки.

Научная новизна работы заключается в том, что: впервые предложен комплекс критериев индивидуальной и групповой информативности экземпляров выборки, позволяющий оценивать качество выборок, что позволяет автоматизировать процесс сравнения выборок между собой; получили дальнейшее развитие методы полного и сокращенного перебора, которые модифицированы путем введения разработанных критериев для оценивания пригодности экземпляров улучшать или ухудшать решения на основе индивидуальной и групповой информативностей, что позволяет автоматизировать процесс формирования репрезентативных выборок по отношению к исходной выборке.

Практическая ценность работы состоит в том, что: разработано программное обеспечение, реализующее предложенные критерии и методы; в результате проведенных экспериментов показана практическая пригодность разработанных критериев для отбора экземпляров; определены оценки сложности разработанных методов, позволяющие определить условия их применимости на практике.

Работа выполнена в рамках госбюджетной темы кафедры программных средств Запорожского национального технического университета “Информационные технологии автоматизации распознавания образов и принятия решений для диагностики в условиях неопределенности на основе гибридных нечеткологических, нейросетевых и мультиагентных методов вычислительного интеллекта” (номер государственной регистрации 0109U007673).

Список литературы: 1. Дубровин, В.И. *Интеллектуальные средства диагностики и прогнозирования надежности авиадвигателей* [Текст] : Монография / В. И. Дубровин, С. А. Субботин, А. В. Богуслаев, В. К. Яценко. – Запорожье: ОАО “Мотор–Сич”, 2003. – 279 с. 2. Субботин, С. О. *Неітеративні, еволюційні та мультиагентні методи синтезу нечіткологічних і нейромережних моделей* [Текст] : Монографія / С. О. Субботін, А. О. Олійник, О. О. Олійник ; під заг.ред. С. О. Субботіна. – Запоріжжя: ЗНТУ, 2009. – 375 с. 3. Джессен, Р. Дж. *Методы статистических обследований* [Текст] / Р.Д. Джессен; пер. с англ. Ю. П. Лукашина, Я. Ш. Паппэ; под ред. Е.М. Четыркина. – М.: Финансы и статистика, 1985. – 478 с. 4. Bernard, H. R. *Social research methods: qualitative and quantitative approaches* [Text] / H. R. Bernard. – Thousand Oaks: Sage Publications, 2006. – 784 p. 5. Кокрен, У. *Методы выборочного исследования* [Текст] / У. Кокрен; пер. с англ. И. М. Соннна; под ред. А. Г. Волкова, Н. К. Дружинина. – М.: Статистика, 1976. – 440 с.

Поступила в редколлегию 26.02.2010 г.

УДК 004.93

Критерії індивідуальної інформативності та методи відбору екземплярів для побудови діагностичних і розпізнавальних моделей / С.А. Суботін // Біоніка інтелекту: наук.-техн. журнал. – 2010. – № 1 (72). – С. 38–42.

З метою автоматизації процесу побудови діагностичних і розпізнавальних моделей запропоновано комплекс критеріїв індивідуальної та групової інформативності екземплярів, а також методи формування вибірки, що дозволяють оцінювати якість, а також автоматизувати процес формування репрезентативних вибірок відносно до вихідної вибірки.

Л. 1. Бібліогр.: 5 найм.

UDC 004.93

Criteria for individual informativity and sampling methods for diagnostic and recognition model building / S.A. Subbotin // Bionics of Intelligence: Sci. Mag. – 2010. – № 1 (72). – P. 38–42.

The set of criteria of individual and group informativity of exemplars, and sampling methods are proposed with the aim to automate the process of diagnostic and recognition model building. It allows to measure quality, and to automate the process of a sample formation representative to the original sample.

Fig. 1. Ref.: 5 items.

УДК 519.86:347.464



ИНТЕРВАЛЬНОЕ ОЦЕНИВАНИЕ ПАРАМЕТРОВ ДЛЯ ОПРЕДЕЛЕНИЯ НАДЕЖНОСТИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

В.Ю. Дубницкий¹, А.М. Кобылин², О.А. Кобылин³

¹ХИБД УБД, г. Харьков, Украина, valeriy_dubn@mail.ru;

²ХИБД УБД, г. Харьков, Украина, kobilin@khibs.edu.ua;

³ХНУРЭ, м. Харьков, Украина, kblin@kture.kharkov.ua;

Предложен способ оценивания параметров, определяющих надёжность программного обеспечения. Для этого выполнено исследование и решение простейших алгебраических уравнений, коэффициенты которых принадлежат множеству интервальных чисел. Разработанное программное средство позволяет пользователю проводить вычисления, выбирая из соответствующих списков названия переменных, совпадающих с названиями показателей, вводить исходные данные, выбирать тип операции, выполнять расчеты и, в случае потребности, сохранять их на листе электронной таблицы и представлять результаты в графическом виде.

ИНТЕРВАЛЬНЫЕ ЧИСЛА, НАДЕЖНОСТЬ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, ОБРАТНАЯ ПОЛЬСКАЯ ЗАПИСЬ, СТЕКОВЫЙ ИНТЕРВАЛЬНЫЙ КАЛЬКУЛЯТОР, МОДЕЛЬ ХОЛСТЕДА, МОДЕЛИ МИЛСА

Введение

Одной из существенных задач в разработке программного обеспечения является его тестирование на наличие ошибок (так называемых багов). Это обусловлено тем, что на качество программного продукта может влиять достаточно много факторов. Они варьируют в зависимости от назначения программного продукта, сложности поставленной задачи и объёмности. Очень часто их характеристики имеют высокий уровень нестохастической неопределённости. Высокая надёжность, предъявляемая к программным системам критического применения, требует проведения большого числа испытаний для достоверного определения их безотказной работы. В [1] приведена оценка необходимого количества испытаний, которая представлена в виде:

$$n = \frac{\lg(1 - \beta)}{\lg(1 - p)}, \quad (1)$$

где: n – необходимое количество испытаний; β – доверительная вероятность, p – верхнее допустимое значение вероятности отказа.

Результаты соответствующих вычислений приведены в табл. 1.

Таблица 1

Определение количества испытаний программной системы при заданной доверительной вероятности и вероятности отказа

Вероятность отказа p	Уровень доверительной вероятности β			
	0,95	0,975	0,99	0,995
$1 \cdot 10^{-2}$	299	367	458	528
$1 \cdot 10^{-3}$	2995	3687	4603	5296
$1 \cdot 10^{-4}$	29956	36887	46049	52981
$1 \cdot 10^{-5}$	999572	368886	460515	529829
$1 \cdot 10^{-6}$	2995731	3688878	4605158	5298315

Из этой таблицы видно, что при испытании программных систем в режиме реального времени и при высоких требованиях к их безотказности количество испытаний достигает сотен тысяч и даже миллионов, а длительность испытаний может быть соизмеримой с месяцами или даже годами.

В этом случае адекватным математическим аппаратом для количественного анализа результатов тестирования компьютерных программ служит аппарат интервальных вычислений [2, 3].

Целью работы является разработка метода, позволяющего формулировать требования к параметрам, определяющим надёжность программного обеспечения при нестохастической их неопределённости. Подобные ситуации имеют место на различных стадиях предварительного проектирования сложных программных продуктов.

1. Анализ предметной области и постановка задачи

В связи с необходимостью решения обратной задачи – подбора параметров программы, гарантирующих ее необходимую надёжность, далее используем систему аксиом АК₂, приведенную в работе [3].

Пусть символ « o » означает одну из операций +, -, *, / имеющих традиционный смысл. Используя алгебраическую символику запишем, что

$$o \in \{+, -, *, /\}. \quad (2)$$

Тогда для двух интервальных чисел $[A_1], [A_2]$ справедливо условие:

$$[A_1] o [A_2] = (\min U, \max U), \quad (3)$$

где

$$U = ((a_1, a_2) o (b_1, b_2)) = (a_1 o b_1), (a_1 o b_2), (a_2 o b_1), (a_2 o b_2). \quad (4)$$

Сравним выполнение операции $(-)$ в системах аксиом AK_1 , приведенной в [2], и AK_2 , приведенной в [3], используя геометрическую иллюстрацию (рис. 1).

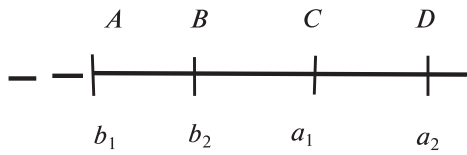


Рис. 1. Геометрическая иллюстрация выполнения операции $(-)$

Очевидно, что:

$$DA = a_2 - b_1; CA = a_1 - b_1; DB = a_2 - b_2; CB = a_1 - b_2.$$

Тогда отрезком наибольшей длины будет отрезок DA, наименьшей – отрезок CB. Таким образом, показано совпадение этих двух систем аксиом, потому что условия (2, 3, 4) совпадают с условием (1).

Следуя работе [3], назовем интервалы $[A] = (a_1, a_2)$ и $[\bar{A}] = (a_2, a_1)$ сопряженными.

В этой же работе показано, что решения алгебраических уравнений, коэффициенты которых есть интервальные числа, могут быть получены в следующем виде:

Уравнение I типа:

$$[A] + [X] = [B], \tag{5}$$

тогда

$$[X] = [B] - [A]. \tag{6}$$

Уравнение II типа:

$$[A] \cdot [X] = [B], 0 \notin [A], \tag{7}$$

тогда

$$[X] = \frac{[B]}{[A]}. \tag{8}$$

Уравнение III типа:

$$[A] [X] + [B] = [C], \tag{9}$$

тогда

$$[X] = \frac{[C] - [\bar{B}]}{[A]} \tag{10}$$

при условии, что $0 \notin [A]$.

Приведем численные примеры.

Рассмотрим решение уравнения типа I.

$$\begin{aligned} [X] + [5; 7] &= [14; 18] \\ [X] &= [14; 18] - [7; 5] \\ U &= (7; 11; 9; 13) \\ \min U &= 7, \max U = 13 \\ [Y] &= [7, 13]. \end{aligned}$$

Рассмотрим решение уравнения типа II.

$$[X] [5; 7] = [14; 18].$$

$$[X] = \frac{[14; 18]}{[7; 5]};$$

$$U = \left(2; \frac{14}{5}; \frac{18}{7}; \frac{18}{5}\right), \min U = 2; \max U = \frac{18}{5}$$

$$[Y] = \left[2; \frac{18}{5}\right].$$

Рассмотрим решение уравнения типа III.

Пусть $[A] = [2; 5]$; $[B] = [4; 9]$; $[C] = [15; 23]$.

Тогда:

$$[2; 5] * [X] + [4; 9] = [15; 23]$$

В соответствии с (28)

$$[X] = \frac{[15; 23] - [9; 4]}{[5; 2]} = \left[\frac{6}{5}; \frac{19}{2}\right].$$

Следовательно, при решении прямых задач применение аксиом AK_1 и AK_2 приводит к одинаковым результатам, при решении обратных задач следует использовать систему аксиом AK_2 .

2. Модели оценки надежности программного обеспечения

Далее рассмотрим основные модели оценки надежности программного обеспечения, приведенные в работе [4], но для их анализа используем аппарат интервальных вычислений.

Модель Холстеда

$$N_{\text{ошибок}} = \frac{V}{E_{\text{критическое}}}, \tag{11}$$

где: V – объем программы; $E_{\text{критическое}}$ – эмпирическая постоянная $E = 2 \cdot 10^3 \dots 5 \cdot 10^3$; $N_{\text{ошибок}}$ – количество ошибок в программе.

В интервальном виде эта модель имеет вид:

$$[N_{\text{ошибок}}] = [A] [V], \tag{12}$$

где

$$[N_{\text{ошибок}}] = [2 \cdot 10^{-4}; 5 \cdot 10^{-4}] [V], \tag{13}$$

то есть получено уравнение вида I.

На рис. 2 приведена Web-форма решения обратной задачи оценки надежности программного обеспечения с использованием интервального калькулятора модели Холстеда.

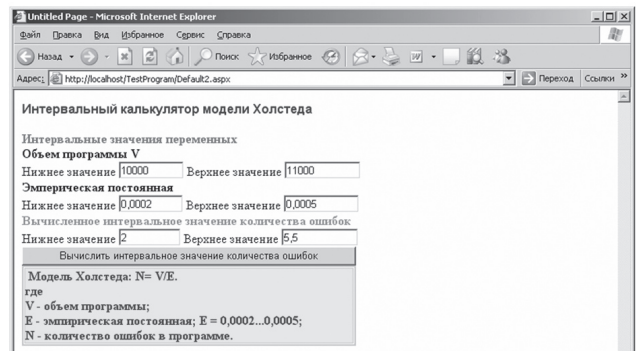


Рис. 2. Web-форма решения обратной задачи оценки надежности программного обеспечения с использованием интервального калькулятора модели Холстеда

Модель Миллса

$$N_{\text{ош}} = \frac{N_{\text{отм}} - N_{\text{внес}}}{N_{\text{внесв}}}, \quad (14)$$

где: $N_{\text{ош}}$ – количество собственных дефектов в программе; $N_{\text{отм}}$ – количество выявленных собственных дефектов в программе; $N_{\text{внес}}$ – количество внесенных дефектов; $N_{\text{внесв}}$ – количество выявленных внесенных дефектов.

Для решения обратной задачи возможны три варианта: решение относительно $N_{\text{отм}}$, относительно $N_{\text{внес}}$, относительно $N_{\text{внесв}}$. Однако в любом случае задача сводится к уравнению вида II. Его варианты приведены в табл. 2.

Таблица 2

Условия задачи по определению параметров модели Миллса

[A]	[X]	[B]
$[N_{\text{отм}}] / [N_{\text{внесв}}]$	$[N_{\text{внес}}]$	$[N_{\text{ош}}]$
$[N_{\text{внес}}] / [N_{\text{внесв}}]$	$[N_{\text{отм}}]$	$[N_{\text{ош}}]$
$[N_{\text{отм}}] / [N_{\text{внес}}]$	$[X] = [1; 1] [N_{\text{внес}}]$ $[N_{\text{внес}}] = [X]^{-1}$	$[N_{\text{ош}}]$

На рис. 3 показана Web-форма решения прямой и обратной задач оценки надежности программного обеспечения с использованием интервального калькулятора модели Миллса.

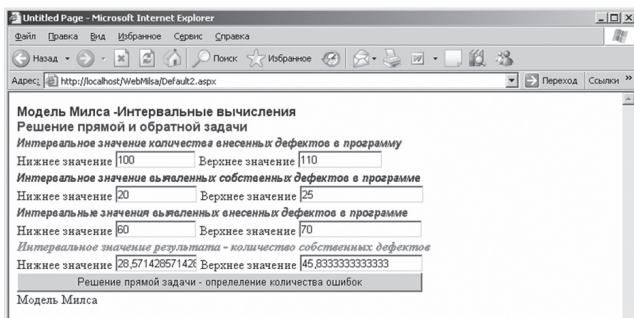


Рис. 3. Web-форма решения прямой и обратной задач оценки надежности программного обеспечения с использованием интервального калькулятора модели Миллса

Модель фирмы IBM

$$N = \alpha \cdot MUM + \gamma \cdot UM, \quad (15)$$

где: α, γ – эмпирические постоянные, $\alpha = 23; \gamma = 2; MUM$ – количество многократно исправляемых модулей (более десяти раз); UM – количество исправляемых модулей (не более десяти раз).

В свою очередь

$$UM = 0,9 NM + 0,15 CUM, \quad (16)$$

$$MUM = 0,15 UM + 0,06 CUM, \quad (17)$$

где NM – количество модулей, CUM – количество старых исправляемых модулей.

Подставив (16) и (17) в (15), в интервальном виде получим уравнение

$$N = [3,45; 3,45] * [MUM] + [1,8; 1,8] * [NM] + [1,68; 1,68] * [CUM]. \quad (18)$$

Пусть

$$[3,45; 3,45] = [F]; \quad [1,8; 1,8] = [G]; \\ [1,68; 1,68] = [Q].$$

Тогда

$$[N] = [F] * [NUM] + [G] * [NM] + [Q] * [CUM]. \quad (19)$$

При решении этого уравнения относительно одного из параметров получим уравнение типа III. Варианты постановки задачи приведены в табл. 3.

Таблица 3

Условия задачи по определению параметров модели фирмы IBM

[A]	[X]	[B]	[C]
[A]	[M]	$[B] * [NM] + [C] * [CUM]$	[N]
[B]	[NM]	$[A] * [M] + [C] * [CUM]$	[N]
[C]	[CUM]	$[A] * [M] + [B] * [NM]$	[N]

3. Результаты исследования

Программирование калькулятора в режиме обратной польской записи является существенным методом повышения надежности программ является сокращение ее объем, в частности, в соответствии с метрикой Холстеда, количество ошибок в программе после окончательной ее разработки определяется по формуле (11). Доказательство данного метода приведено в работе [5]. Для сокращения объема программ предлагается использование так называемой обратной польской записи (ОПЗ). Сравнительный анализ объема программ, разработанных в обычной нотации и ОПЗ, показал, что для формул, в состав которых входит более 5 операндов, экономия составляет примерно 17%.

Для реализации ОПЗ использовался стек, являющийся наиболее важной из структур данных. Эти структуры встречаются в программировании буквально на каждом шагу, в самых разнообразных ситуациях. Особенно интересен стек, который имеет самые неожиданные применения. В свое время при разработке серии ЭВМ IBM 360 в начале 70-х годов XX века фирма IBM совершила драматическую ошибку, не предусмотрев аппаратную реализацию стека. Эта серия содержала много других неудачных решений, но, к сожалению, была скопирована в Советском Союзе под названием ЕС ЭВМ (Единая Серия), а все собственные разработки были приостановлены. Это отбросило советскую промышленность на много лет назад в области разработки компьютеров.

Стек – самая популярная и, пожалуй, самая важная структура данных в программировании. Стек представляет собой запоминающее устройство, из которого элементы извлекаются в порядке,

обратном их добавлению. Это как бы неправильная очередь, в которой первым обслуживают того, кто встал в нее последним. В программистской литературе общепринятыми являются аббревиатуры, обозначающие дисциплину работы стека. Дисциплина работы стека обозначается LIFO, последним пришел — первым уйдешь (Last In First Out).

Стек можно представить в виде трубки с подпружиненным дном, расположенной вертикально. Верхний конец трубки открыт, в него можно добавлять, или, как говорят, заталкивать элементы. Общепринятые английские термины в этом плане очень красочны, операция добавления элемента в стек обозначается push, в переводе “затолкнуть, запихнуть”. Новый добавляемый элемент проталкивает элементы, помещенные в стек ранее, на одну позицию вниз. При извлечении элементов из стека они как бы выталкиваются вверх, по-английски pop (“выстреливают”).

Примером стека может служить стопка бумаг на столе, стопка тарелок и тому подобное. Отсюда произошло название стека, что по-английски означает стопка. Тарелки снимаются со стопки в порядке, обратном их добавлению. Доступна только верхняя тарелка, то есть тарелка на вершине стека. Хорошим примером будет также служить железнодорожный тупик, в который можно составлять вагоны.

Стек применяется довольно часто, причем в самых разных ситуациях. Объединяет их следующая цель: нужно сохранить некоторую работу, которая еще не выполнена до конца, при необходимости переключения на другую задачу. Стек используется для временного сохранения состояния не выполненного до конца задания. После сохранения состояния компьютер переключается на другую задачу. По окончании ее выполнения состояние отложенного задания восстанавливается из стека, и компьютер продолжает прерванную работу.

Почему именно стек используется для сохранения состояния прерванного задания? Предположим, что компьютер выполняет задачу *A*. В процессе ее выполнения возникает необходимость выполнить задачу *B*. Состояние задачи *A* запоминается, и компьютер переходит к выполнению задачи *B*. Но ведь и при выполнении задачи *B* компьютер может переключиться на другую задачу *C*, и нужно будет сохранить состояние задачи *B*, прежде чем перейти к *C*. Позже, по окончании *C* будет вначале восстановлено состояние задачи *B*, затем, по окончании *B* — состояние задачи *A*. Таким образом, восстановление происходит в порядке, обратном сохранению, что соответствует дисциплине работы стека.

Стек позволяет организовать рекурсию, то есть обращение подпрограммы к самой себе либо непосредственно, либо через цепочку других вызо-

вов. Пусть, например, подпрограмма *A* выполняет алгоритм, зависящий от входного параметра *X* и, возможно, от состояния глобальных данных. Для самых простых значений *X* алгоритм реализуется непосредственно. В случае более сложных значений *X* алгоритм реализуется как сведение к применению того же алгоритма для более простых значений *X*. При этом подпрограмма *A* обращается сама к себе, передавая в качестве параметра более простое значение *X*. При таком обращении предыдущее значение параметра *X*, а также все локальные переменные подпрограммы *A* сохраняются в стеке. Далее создается новый набор локальных переменных и переменная, содержащая новое (более простое) значение параметра *X*. Вызванная подпрограмма *A* работает с новым набором переменных, не разрушая предыдущего набора. По окончании вызова старый набор локальных переменных и старое состояние входного параметра *X* восстанавливаются из стека, и подпрограмма продолжает работу с того места, где она была прервана.

На самом деле даже не приходится специальным образом сохранять значения локальных переменных подпрограммы в стеке. Дело в том, что локальные переменные подпрограммы (то есть ее внутренние, рабочие переменные, которые создаются в начале ее выполнения и уничтожаются в конце) размещаются в стеке, реализованном аппаратно на базе обычной оперативной памяти. В самом начале работы подпрограмма захватывает место в стеке под свои локальные переменные, этот участок памяти в аппаратном стеке называют обычно блок локальных переменных или по-английски frame (“кадр”). В момент окончания работы подпрограмма освобождает память, удаляя из стека блок своих локальных переменных.

Кроме локальных переменных, в аппаратном стеке сохраняются адреса возврата при вызовах подпрограмм. Пусть в некоторой точке программы *A* вызывается подпрограмма *B*. Перед вызовом подпрограммы *B* адрес инструкции, следующей за инструкцией вызова *B*, сохраняется в стеке. Это так называемый адрес возврата в программу *A*. По окончании работы подпрограмма *B* извлекает из стека адрес возврата в программу *A* и возвращает управление по этому адресу. Таким образом, компьютер продолжает выполнение программы *A*, начиная с инструкции, следующей за инструкцией вызова. В большинстве процессоров имеются специальные команды, поддерживающие вызов подпрограммы с предварительным помещением адреса возврата в стек и возврат из подпрограммы по адресу, извлекаемому из стека. Обычно команда вызова называется call, команда возврата — return.

В стек помещаются также параметры подпрограммы или функции перед ее вызовом. Порядок их помещения в стек зависит от соглашений, при-

нятых в языках высокого уровня. Так, в языке Си или C++ на вершине стека лежит первый аргумент функции, под ним второй и так далее. В Паскале всё наоборот, на вершине стека лежит последний аргумент функции. (Поэтому, кстати, в Си возможны функции с переменным числом аргументов, такие, как printf, а в Паскале нет.). В языке C# для работы со стеком создан специальный класс Stack, свойства и методы которого представлены в табл. 4 и 5.

Таблица 4

Свойство класса Stack

Имя	Описание
Count	Возвращает число элементов в стеке

Таблица 5

Методы Stack

Имя	Описание
Pop	Возвращает элемент с вершины стека, одновременно удаляя его
Push	Добавляет элемент на вершину стека
Peek	Возвращает верхний элемент стека, не удаляя его

В работе рассматривается создание специализированного интервального калькулятора, реализующего обратную польскую запись с использованием стека для выполнения финансовых расчетов.

Стековый интервальный калькулятор и обратная польская запись формулы.

В 1920 г. польский математик Ян Лукашевич предложил способ записи арифметических формул, не использующий скобок. В привычной нам записи знак операции записывается между аргументами, например сумма чисел 2 и 3 записывается как 2 + 3. Ян Лукашевич предложил две другие формы записи: префиксная форма, в которой знак операции записывается перед аргументами, и постфиксная форма, в которой знак операции записывается после аргументов. В префиксной форме сумма чисел 2 и 3 записывается как + 2 3, в постфиксной — как 2 3 +. В честь Яна Лукашевича эти формы записи называют прямой и обратной польской записью.

В польской записи скобки не нужны. Например выражение

$$(2+3)*(15-7)$$

записывается в прямой польской записи как

$$* + 2 3 - 15 7,$$

в обратной польской записи — как

$$2 3 + 15 7 - *.$$

В стековом интервальном калькуляторе с ОПЗ для вычисления выражения

$$[2;3]+[3;4]-[1;3]$$

необходимо записать

$$2 3 3 4 + 1 3 -$$

Если прямая польская запись не получила большого распространения, то обратная оказалась чрезвычайно полезной. Неформально преимущество обратной записи перед прямой польской записью или обычной записью можно объяснить тем, что гораздо удобнее выполнять некоторое действие, когда объекты, над которыми оно должно быть совершено, уже даны.

Обратная польская запись формулы позволяет вычислять выражение любой сложности, используя стек как запоминающее устройство для хранения промежуточных результатов. Обычные модели калькуляторов не позволяют вычислять сложные формулы без использования бумаги и ручки для записи промежуточных результатов. В некоторых моделях есть скобки с одним или двумя уровнями вложенности, но более сложные выражения вычислять невозможно. Также в обычных калькуляторах трудно понять, как результат и аргументы перемешаются в процессе ввода и вычисления между регистрами калькулятора. Калькулятор обычно имеет регистры X, Y и регистр памяти, промежуточные результаты каким-то образом перемешаются по регистрам, каким именно — запомнить невозможно.

В отличие от других калькуляторов, устройство стекового калькулятора вполне понятно и легко запоминается. Калькулятор имеет память в виде стека. При вводе числа оно просто добавляется в стек. При нажатии на клавишу операции, например на клавишу +, аргументы операции сначала извлекаются из стека, затем с ними выполняется операция, наконец, результат операции помещается обратно в стек. Таким образом, при выполнении операции с двумя аргументами, например сложения, в стеке должно быть не менее двух чисел. Аргументы удаляются из стека и на их место записывается результат, то есть при выполнении сложения глубина стека уменьшается на единицу. Вершина стека всегда содержит результат последней операции и высвечивается на дисплее калькулятора.

Реализация стекового интервального калькулятора на C#.

Рассмотрим проект, реализующий стековый интервальный калькулятор на C#. Такая программа весьма полезна, поскольку позволяет проводить вычисления, не прибегая к записи промежуточных результатов на бумаге.

Программа состоит из трех файлов: "Form1.cs", "Form2.cs" и "Program.cs". Первые два файла реализуют стек вещественных чисел, эта реализация уже рассматривалась ранее. Файл "Form1.cs" реализует интервальный стековый калькулятор на базе ОПЗ. Файл Form2.cs реализует финансовые вычисления структур данных типа стек на ОПЗ. Файл Program.cs — основной файл проекта.

Ниже приведено содержимое двух фрагментов файла "Form1.cs". Функция main, описанная в этом файле, организует диалог с пользователем в режиме команда-ответ. Пользователь может ввести число с клавиатуры, это число просто добавляется в стек. При вводе одного из четырех знаков арифметических операций +, -, *, / программа извлекает из стека два числа, выполняет указанное арифметическое действие над ними и помещает результат обратно в стек. Значение результата отображается также на дисплее. Кроме арифметических операций, пользователь может ввести название одной из стандартных функций: exp(x), ln, log (десятичный логарифм), sqrt(x). При этом программа извлекает из стека аргумент функции, вычисляет значение функции и помещает его обратно в стек. При желании список стандартных функций и возможных операций можно расширить. Каждая команда стекового калькулятора реализуется с помощью отдельной функции. Например сложение реализуется с помощью функции AddIntOPN():

```
private void AddIntOPN_Click(object sender,
EventArgs e)
{
    if (EntryInProgress)
        InitializeRegisterFromDisplay();
    if (RegStack1.Count >= 2)
    {
        double op4 = (double)RegStack1.Pop();
        double op3 = (double)RegStack1.Pop();
        double op2 = (double)RegStack1.Pop();
        double op1 = (double)RegStack1.Pop();
        double a = op1 + op3;
        double b = op2 + op4;
        string display1 = a.ToString(FormatString);
        string display2 = b.ToString(FormatString);
        textBox3.Text = display1;
        textBox4.Text = display2;
        RegStack1.Push(a);
        RegStack1.Push(b);
        Display1.Text = " ";
        Display1.Text = " ";
    }
}
```

В начале функции проверяется, что глубина стека не меньше двух. В противном случае, выдается сообщение об ошибке, и функция завершается. Далее из стека извлекаются операнды y и x операции вычитания. Элементы извлекаются из стека в порядке, обратном их помещению в стек, поэтому y извлекается раньше, чем x. Затем вычисляется разность y - x, ее значение помещается обратно в стек и печатается на дисплее, для печати вершины стека вызывается функция display.

Приведем начальный фрагмент программы, описывающий объявление экземпляров класса Stack и переменных.

```
namespace Калькулятор_Интервальный_ОПЗ
{
    public partial class IntSQRT : Form
    {
        public IntSQRT()
        {
            InitializeComponent();

            Stack RegStack1 = new Stack();
            Stack RegStack2 = new Stack();
            string FormatString = "f2";
            const int MaxChars = 21;
            private bool FixPending = false;
            private bool DecimalInString = false;
            private bool EntryInProgress = false;
            private void
                InitializeRegisterFromDisplay()
            {
                double x = (Display1.Text.Length == 0 || Display1.
                Text == ",") ? 0.0 : Convert.ToDouble(Display1.
                Text);
                RegStack1.Push(x);
                double x1 = (Display2.Text.Length == 0 || Display2.
                Text == ",") ? 0.0 : Convert.ToDouble(Display2.
                Text);
                RegStack1.Push(x1);
            }
        }
    }
}
```

В табл. 6 и 7 приведены сравнительные результаты для выражений различной степени сложности на обычном калькуляторе, калькуляторе с ОПЗ и интервальном калькуляторе с ОПЗ.

При выполнении расчетов для начисления простых и сложных процентов, количество операций сократилось на две. При выполнении расчетов на интервальном стековом калькуляторе количество операций сократилось на три. Таким образом с повышением сложности расчетов эффект от использования ОПЗ увеличивается.

Таблица 6

Формулы начисления простых и сложных процентов

Обычный калькулятор			
Действия на обычном калькуляторе		Действия на калькуляторе с ОПЗ	
$S=P(1+rt)$	$S=P(1+r)^t$	$S=P(1+rt)$	$S=P(1+r)^t$
0,1	1	0,1	0,1
*	+	Ввод	Ввод
5	0,1	5	1
=	=	*	+
+	x^y	1	5
1	5	+	a^x
=	=	1000	1000
*	*	*	*(1610,51)
1000	1000		
=	=(1610,51)		

Таблица 7

Формулы начисления простых и сложных процентов

Интервальный калькулятор	
Действия на интервальном калькуляторе с ОПЗ	
$[S_n; S_k] = [P_n; P_k] \cdot (1 + [r_n; r_k] \cdot [t_n; t_k])$	$[S_n; S_k] = [P_n; P_k] \cdot (1 + [r_n; r_k])^{[t_n; t_k]}$
[0,1; 0,1] Ввод	[1;1] Ввод
[5,0; 5,1] Ввод	[0,1;0,1] Ввод
Умножение интервальное [1;1] Ввод	Сложение интервальное [5;5,1] Ввод
Сложение интервальное [1000;1100] Ввод	Возведение в степень интервальное a^x [1000;1100] Ввод
Умножение интервальное [1500;1717,1]	Умножение интервальное [1610,51;1873,01]

S – накопленная сумма, P – сумма вклада, r – процентная ставка, t – срок вклада.

$$P = 1000, r = 0,1, t = 5.$$

S – накопленная сумма ренты, r – размер члена ренты, i – годовая процентная ставка, n – срок ренты (в годах)

Выводы

Показана эффективность применения интервальных вычислений для решения задач, связанных с тестированием программ в условиях, в которых применение традиционных методов прогнозирования невозможно или усложнено отсутствием сведений о статистических свойствах переменных.

Приведены сведения о специализированном программном стековом калькуляторе, который реализует правила интервальной арифметики для оценивания эффективности тестирования компьютерных программ.

Показано, что применение обратной польской записи позволяет сократить объем программы, следовательно, повысить надежность.

В связи с увеличением объема финансовых вычислений и в целях повышения ответственности за их результаты к перспективным направлениям

дальнейших работ следует отнести организацию схемы дистанционного доступа к вычислительным ресурсам и создание специализированных сетей, предназначенных для анализа финансовых данных.

Список литературы: 1. *Вентцель, Е. С.* Теория вероятностей [Текст] : учебник / Е. С. Вентцель. – М. : Гос. изд-во физмат. лит-ры, 1962. – 560 с. 2. *Алефельд, Г.* Введение в интервальные вычисления [Текст] / Г. Алефельд, Ю. Херцбергер; – М.: Мир, 1987. – 259 с. 3. *Алтунин, А. Е.* Модели и алгоритмы принятия решений в нечетких условиях [Текст] / А. Е. Алтунин, М. В. Семухин; – Тюмень: Изд. ТГУ, 2000. – 352 с. 4. *Харченко, В. С.* Методы моделирования и оценки качества и надежности программного обеспечения [Текст] / В. С. Харченко, В. В. Скляр, О. М. Тарасюк; – Учеб. пособие. – Харьков: Нац. аэрокосм. ун-т “Харьк. авиац. ин-т”, 2004. – 159 с. 5. *Пайчун, Б. П.* Оценка надежности программного обеспечения [Текст] / Б. П. Пайчун, Р. М. Юсупов; – Спб.: Наука, 1994. – 84 с.

Поступила в редколлегию 02.03.2010 г.

УДК 519.86:347.464

Інтервальні оцінювання параметрів для визначення надійності програмного забезпечення / В.Ю. Дубницький, А.М. Кобилін, О.А. Кобилін // Біоніка інтелекту: наук.-техн. журнал. – 2010. – № 1 (72). – С. 43–49.

Показано ефективність застосування інтервальних обчислень для розв’язання задач, пов’язаних з тестуванням програм. Показано, що застосування зворотного польського запису дозволяє скоротити обсяг програми, отже, підвищити надійність.

Табл. 7. Іл. 3. Бібліогр.: 5 найм.

UDK 519.86:347.464

Interval evaluation of parameters to determine software reliability / V.Y. Dubnickiy, A.M. Kobylin, O.A. Kobylin // Bionics of Intelligence: Sci. Mag. – 2010. – № 1 (72). – P. 43–49.

Efficiency of interval calculations application for solving the tasks connected to the programs testing is shown. It is also proved that using the backwards polish notation allows to reduce the volume of the program, and so to improve its reliability.

Table 7. Fig. 3. Ref.: 5 items.

УДК 519.688:004.896



ИДЕНТИФИКАЦИЯ ЧАСТНОЙ ПОЛЕЗНОСТИ МНОГОФАКТОРНЫХ АЛЬТЕРНАТИВ С ПОМОЩЬЮ S-ОБРАЗНЫХ ФУНКЦИЙ

В.В. Бескоровайный¹, Е.В. Соболева²

¹ ХНУРЭ, г. Харьков, Украина, beskorovainyi@kture.kharkov.ua;

² ХНУРЭ, г. Харьков, Украина, se_@ukr.net

Предложены модификации S-образных функций полезности частных критериев, используемых в задачах многокритериального оценивания и выбора в рамках кардиналистического подхода. Приведены результаты экспериментальных исследований временной сложности и точности процедур аппроксимации частной полезности на основе модифицированных функций Гаусса и логистической функции.

ПРИНЯТИЕ РЕШЕНИЙ, МНОГОКРИТЕРИАЛЬНОЕ ОЦЕНИВАНИЕ, ФУНКЦИЯ ПОЛЕЗНОСТИ ЧАСТНЫХ КРИТЕРИЕВ, ИДЕНТИФИКАЦИЯ, МОДЕЛЬ, МЕТОД, АЛГОРИТМ

Введение

Одной из важнейших задач бионики интеллекта является изучение процессов оценивания и выбора решений человеком. Формализация и моделирование этих процессов позволят совершенствовать существующие и создавать новые, более эффективные системы поддержки принятия решений для систем проектирования и управления. Подавляющее большинство задач, возникающих в процессах принятия решений, по своей сути являются многокритериальными. В рамках кардиналистического подхода для выбора наилучшего компромиссного решения используется парадигма максимизации полезности [1]. Лицо, принимающее решение (ЛПР), при выборе вариантов из множества допустимых приписывает им некоторую полезность (ценность), количественные значения которой и определяют его выбор. Для количественной оценки предпочтений ЛПР требуется определение метрики в виде функции обобщенной полезности, позволяющей ранжировать альтернативы.

1. Анализ существующих моделей частной полезности

Функции общей полезности строятся на основе линейных или нелинейных функций полезности частных критериев (ФПЧК) [2-6]. При этом к ФПЧК предъявляется ряд требований [1]: монотонность, безразмерность, диапазон изменения [0, 1], инвариантность к виду экстремума, возможность реализации линейных и характерных нелинейных зависимостей от значения частного критерия. Кроме того, желательно, чтобы функции полезности всех частных критериев имели один и тот же вид и различались только значениями параметров, что позволило бы свести задачи их формирования для всех частных критериев к решению задач выбора наилучших значений их параметров. Наибольшее применение на практике находит ФПЧК вида [1]:

$$\xi(x) = \bar{x}^\alpha = \left(\frac{x - x_{\text{нл}}}{x_{\text{нл}} - x_{\text{нх}}} \right)^\alpha, \quad (1)$$

где $x, x_{\text{нл}}, x_{\text{нх}}, \bar{x}$ – соответственно исходное (текущее), наилучшее, наихудшее и нормированное значения частного критерия; α – параметр, определяющий конкретный вид зависимости (линейная выпуклая, вогнутая). Зависимости, отражающие изменение эффективности большинства антропогенных объектов, являются S-образными и представляются:

– функцией Гаусса [3]:

$$\xi(x) = e^{-\frac{(\bar{x} - \bar{x}_{\text{нл}})^2}{c}}, \quad (2)$$

где $c > 0$ – параметр, определяющий конкретный вид зависимости;

– логистической функцией:

$$\xi(x) = \frac{1}{1 + e^{-\frac{(\bar{x} - a)}{b}}}, \quad (3)$$

где a – абсцисса точки перегиба; b – параметр, определяющий вид зависимости;

– функцией Харрингтона [7]:

$$\xi(x) = e^{-e^{-(k \cdot \bar{x} - a)}}, \quad (4)$$

где k – параметр нелинейности, a/k – определяет точку перегиба;

– функцией-склейкой степенных функций [4]:

$$\xi(x) = \begin{cases} a \cdot \left(\frac{\bar{x} - \bar{x}_{\text{нх}}}{\bar{x}_a - \bar{x}_{\text{нх}}} \right)^{\alpha_1}, & \bar{x}_{\text{нх}} \leq \bar{x} \leq \bar{x}_a, \\ a + (1 - a) \cdot \left(\frac{\bar{x} - \bar{x}_a}{\bar{x}_{\text{нл}} - \bar{x}_a} \right)^{\alpha_2}, & \bar{x}_a \leq \bar{x} \leq \bar{x}_{\text{нл}}, \end{cases} \quad (5)$$

где \bar{x}_a, a – координаты точки перегиба ($0 \leq a \leq 1$); α_1, α_2 – коэффициенты нелинейности, определяющие вид зависимости соответственно на начальном и конечном отрезках.

Приведенные ФПЧК (1) – (4) имеют недостатки, снижающие адекватность моделей многокритериального оценивания и выбора. Характер зависимостей ФПЧК (2) – (4) от значений параметров приведен на рис. 1.

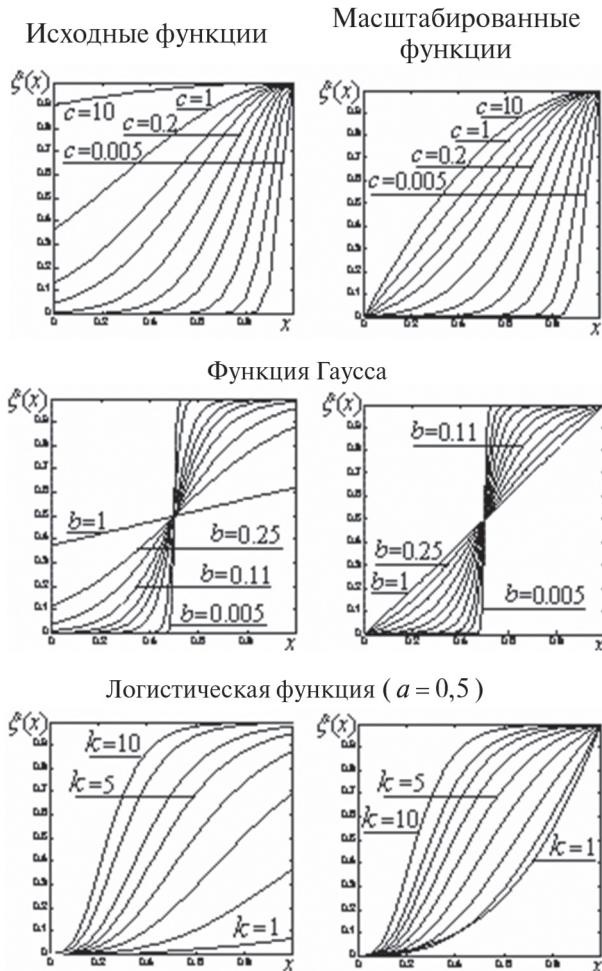


Рис. 1. Базовые S-образные ФПЧК

Основной недостаток ФПЧК (2) – (4) – различная степень приближения значений к 0 и 1 при $x \rightarrow x_{\text{нх}}$ и $x \rightarrow x_{\text{нл}}$, что сокращает их способность к дифференциации полезности различных значений частного критерия. Данный недостаток, при необходимости, преодолевается путем масштабирования ординаты (рис. 1, правая часть):

$$\xi'(x) = \frac{\xi(x) - \xi(x_{\text{нх}})}{\xi(x_{\text{нл}}) - \xi(x_{\text{нх}})}.$$

Изменение параметров функций (2) – (4) существенно не влияет на общий вид представляемых ими зависимостей, изменяя лишь участок, используемый в качестве ФПЧК. Гораздо большее влияние на перемещение точки перегиба и характер зависимостей может оказать введение дополнительных параметров в показатель степени.

Недостатками функции-склейки (5) являются относительно большое количество подлежащих определению параметров и разрыв производной в

точке перегиба. Характер зависимостей ФПЧК (5) для различных значений параметров нелинейности приведен на рис. 2.

ФПЧК (2) – (4) имеют большой угол вхождения в одну или обе зоны нечувствительности и, следовательно, если значения частного критерия будут близки к экстремальным, то погрешность в их определении окажет существенное влияние на погрешность вычисления полезности.

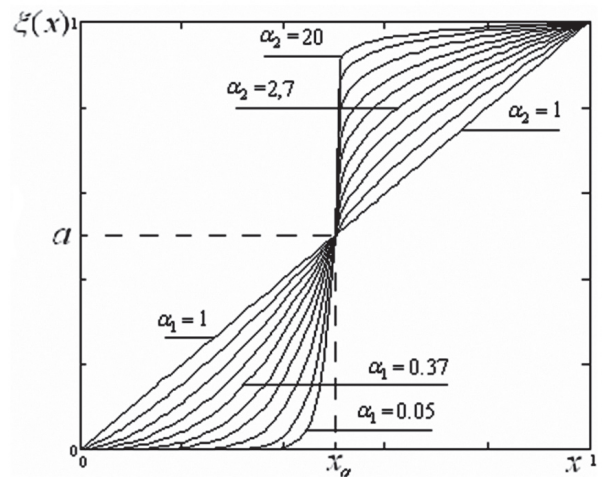


Рис. 2. Семейство ФПЧК (5) при изменении параметров нелинейности

С учетом этого актуальными остаются задачи поиска универсальных ФПЧК, пригодных для оценивания вариантов в различных ситуациях выбора, имеющих минимальное количество параметров, разработки процедур выбора наилучших значений (идентификации) их параметров, имеющих более низкую временную сложность.

Целью статьи является описание синтеза моделей частной полезности многофакторных альтернатив в классе S-образных функций путем решения задачи структурно-параметрической идентификации.

Постановка задачи. Пусть экспертным путем или в результате решения задачи компараторной идентификации [9] определены оценки полезности $(u_i, i = \overline{1, n})$, соответствующие значениям частного критерия x_i . Необходимо в соответствии с заданным критерием подобия k определить вид и параметры A наиболее адекватной модели (ФПЧК): $X, U \xrightarrow{k} \xi^*(X, A^*)$.

2. Модификация ФПЧК на основе функции Гаусса

С увеличением значения c зависимость (2) становится более полой, точка изгиба смещается в сторону $x_{\text{нх}}$, функция становится практически выпуклой, однако по достижении некоторого значения увеличение значения параметра c практически не влияет на форму зависимости (рис. 1).

Для преодоления недостатка ФПЧК (2), связанного с ограниченной возможностью формализа-

ции выпуклых зависимостей, предлагается ввести в нее дополнительный параметр α (рис. 3):

$$\xi(x) = e^{-\frac{(\bar{x}-x_{нл})^{2\alpha}}{c}}, \quad (6)$$

где α – дополнительный параметр, определяющий вид нелинейности.

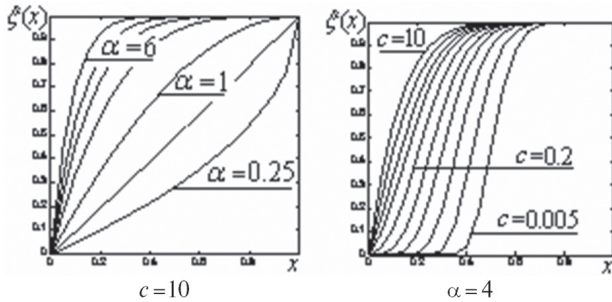


Рис. 3. Модифицированная ФПЧК на основе функции Гаусса (6) со степенным параметром

3. Модификация логистической функции [8]

С учетом того, что прообразом функции (3) является функция гиперболического тангенса

$$f(x) = \frac{1}{2} + \frac{1}{2}th(x) = \frac{1}{2} + \frac{1}{2} \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{1}{1 + e^{-2x}},$$

для устранения ее недостатков предлагается ввести в знаменатель параметры $\alpha, \beta \geq 1$:

$$\xi(x, \alpha, \beta) = \frac{e^x - e^{-x}}{e^{\alpha x} + e^{-\beta x}}. \quad (7)$$

Функция (7), масштабированная от точки минимума $(\bar{x}, \xi(\bar{x}, \alpha, \beta))$ до точки максимума $(\hat{x}, \xi(\hat{x}, \alpha, \beta))$, определяет семейство гладких S-образных функций (рис. 4):

$$\xi'(x, \alpha, \beta) = \frac{\xi(p(x), \alpha, \beta) - \xi(\bar{x}, \alpha, \beta)}{\xi(\hat{x}, \alpha, \beta) - \xi(\bar{x}, \alpha, \beta)}, \quad (8)$$

где $p(x) = (\hat{x} - \bar{x}) \cdot \frac{x - x_{нл}}{x_{нл} - x_{нх}} + \bar{x}$.

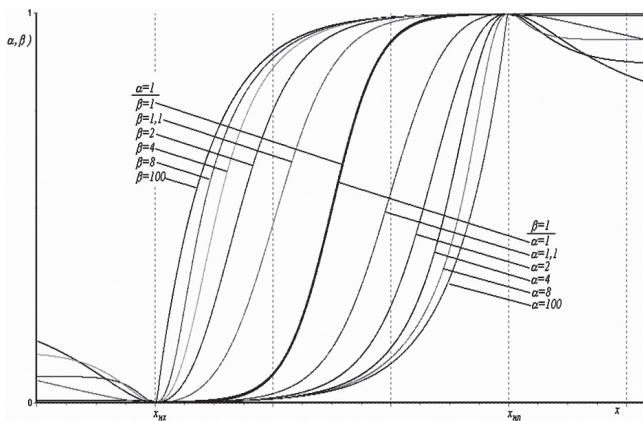


Рис. 4. Семейство масштабированных модифицированных логистических функций (8)

Ввиду сложности аналитического вычисления экстремумов функции (7) (при $z = e^x$

$$\frac{d\xi}{dx} = \frac{(1-\alpha)z^{\alpha+\beta+2} + (1+\alpha)z^{\alpha+\beta} + (1+\beta)z^2 + 1 - \beta}{z^{1+\beta} \cdot (z^\alpha + z^{-\beta})^2}$$

для нахождения экстремумов и масштабирования предлагается программно реализованная процедура численного решения. Экспериментально установлено, что экстремумы функции (7) ограничены значениями:

$$\bar{x} < \frac{1}{2} \ln\left(\frac{b-1}{b+1}\right), \quad \hat{x} > \frac{1}{2} \ln\left(\frac{a+1}{a-1}\right).$$

Предложенная функция (8) отображает все множество действительных чисел на интервал $[0, 1]$. Ее достоинствами являются гладкость, непрерывность и минимальные углы вхождения в нечувствительные зоны, она имеет всего два параметра α и β (которые могут быть представлены как отношение одного к другому α/β или β/α). При этом для эксперта интерпретация параметров предложенной функции может быть следующей: «Во сколько раз рост критерия вблизи наихудшего значения важнее (полезнее) его роста вблизи наилучшего значения?» – в таком случае указанное экспертом соотношение можно использовать непосредственно для оценки параметров как отношение β/α .

4. Результаты экспериментального исследования процедур аппроксимации локальной полезности вариантов

Использование в процессе идентификации частной полезности процедур масштабирования для некоторых ФПЧК затрудняет применение классических методов аппроксимации. Для сравнительного анализа эффективности рассмотренных ФПЧК было разработано программное обеспечение и проведены серии экспериментов.

Имитируя работу экспертов с использованием генератора случайных чисел, формировались по $n-2$ значения частного критерия и соответствующего ему значения полезности. Полученные значения упорядочивались, и к ним добавлялись пары граничных точек $(0,0)$ и $(1,1)$. С помощью последовательного поиска осуществлялась оптимизация параметров ФПЧК по критерию минимума суммы модулей отклонений: $K^0 = \min_j \sum_i |\xi(x_i, A_j) - u_i|$.

Было проведено две серии экспериментов: в каждой серии для каждого значения количества точек $n = 5, 10, 15$ и 20 по 1000 экспериментов. Для модели (8) была также исследована модификация, когда процедура поиска экстремумов заменялась вычислением приближенных оценок:

$$\bar{x}' = \frac{1}{2} \ln\left(\frac{b-1}{b+1}\right); \quad \hat{x}' = -\frac{1}{2} \ln\left(\frac{a-1}{a+1}\right). \quad (9)$$

Анализируемыми показателями качества моделей были: максимальная δK_{\max} и средняя $\delta K_{\text{сред}}$ относительные погрешности ($\delta K = |K^o - \tilde{K}^o| / \tilde{K}^o$, где K^o и \tilde{K}^o – минимальные значение критерия, полученные с помощью данной и наилучшей модели); среднее время решения задачи ($t_{\text{сред}}$); вероятность того, что модель является лучшей (P). Эксперименты проводились на персональной ЭВМ с процессором *Celeron 540* (тактовая частота 1.86 ГГц).

Поиск наилучших значений параметров осуществлялся с одинаковым количеством шагов для каждого из них с осуществлением процедуры поиска экстремумов и с вычислением приближенных оценок экстремумов (9). Наборы значений параметров выбирались таким образом, чтобы множество формируемых кривых равномерно покрывало пространство определения частной полезности. Эксперименты проводились со следующими наборами значений параметров (табл. 1):

– для функции Гаусса (ФГ) (2): $c = [10; 1; 0,5; 0,33; 0,2; 0,12; 0,07; 0,035; 0,015; 0,005]$;

– для модифицированной функции Гаусса (МФГ) (6): $a = 1, 2, \dots, 10$;

– для логистической функции (ЛФ) (3): $a = 0,0,1, \dots, 1$; $b = [1; 0,25; 0,16; 0,11; 0,08; 0,06; 0,04; 0,025; 0,015; 0,005]$;

– для функции Харрингтона (ФХ) (4): $a, k = 1, 2, \dots, 10$;

– для функции-склейки (ФС) (5): $x_a, a = 0, 0,1, \dots, 1$, $\alpha_1, \alpha_2 = [1; 0,779; 0,607; 0,427; 0,368; 0,273; 0,202; 0,135; 0,082; 0,05]$;

– для модифицированной функции гипертангенса (МФГТ) (8) и его упрощенной версии, использующей вычисление оценок (9) (МФГТУ): $\alpha, \beta = [1,01; 1,025; 1,05; 1,1; 1,25; 1,5; 2; 3; 5; 10; 100]$.

Анализ результатов показывает, что функция-склейка дает минимальную погрешность среди всех ФЧПК (в данном случае за счет большого числа вариантов соотношения параметров, то есть большого числа просматриваемых вариантов зависимостей).

Временная сложность параметрической оптимизации рассмотренных ФЧПК с достаточной степенью достоверности аппроксимируется полиномиальными функциями:

– для ФГ (2) – $o(n) = 0,0047n + 0,0156$;

– для МФГ (6) – $o(n) = 0,2n - 0,0078$;

– для ЛФ (3) – $o(n) = -0,0672n + 0,0938$;

– для ФХ (4) – $o(n) = 0,0352n^2 + 0,0039n - 0,2305$;

– для ФС (5) – $o(n) = 0,0078n^2 + 21,811n - 1,8594$;

– для МФГТ (8) –

$o(n) = 0,0352n^2 - 0,1273n + 0,6758$;

– для МФГТУ – $o(n) = 0,0453n + 0,0703$.

Таблица 1

Результаты исследования временной сложности процедур аппроксимации частной полезности

n	Модель	$t_{\text{сред}}, \text{мс}$	P	δK_{\max}	$\delta K_{\text{сред}}$
5	ФГ (2)	0,0156	0,002	1,9410	0,3246
	МФГ (6)	0,1563	0,012	0,9496	0,2179
	ЛФ (3)	0,1094	0,028	1,0453	0,2045
	ФХ (4)	0,2813	0,009	1,8424	0,2117
	ФС (5)	19,9844	0,947	0,1335	0,0016
	МФГТ (8)	0,5625	0,004	1,6186	0,2789
	МФГТУ	0,0938	0,000	2,6961	0,7420
10	ФГ (2)	0,0313	0,002	2,1010	0,4736
	МФГ (6)	0,4219	0,004	1,7454	0,4046
	ЛФ (3)	0,3125	0,010	1,4001	0,2947
	ФХ (4)	0,3438	0,003	1,4627	0,3260
	ФС (5)	41,7188	0,980	0,1013	0,0005
	МФГТ (8)	0,6250	0,003	1,7857	0,5179
	МФГТУ	0,1719	0,000	4,4235	1,4519
15	ФГ (2)	0,0313	0,000	1,9348	0,5671
	МФГ (6)	0,6406	0,001	1,9348	0,5327
	ЛФ (3)	0,2813	0,007	1,4074	0,3227
	ФХ (4)	0,5938	0,006	1,6195	0,3821
	ФС (5)	63,7188	0,984	0,0561	0,0003
	МФГТ (8)	0,5469	0,002	2,1848	0,8027
	МФГТУ	0,2500	0,000	6,6009	2,1604
20	ФГ (2)	0,0313	0,003	2,7813	0,6928
	МФГ (6)	0,7500	0,003	2,7813	0,6693
	ЛФ (3)	0,3438	0,012	1,8431	0,3552
	ФХ (4)	0,7969	0,005	1,8664	0,4423
	ФС (5)	85,4844	0,980	0,2508	0,0007
	МФГТ (8)	0,7500	0,000	2,7083	1,1485
	МФГТУ	0,2188	0,000	8,1802	2,9113

5. Применение модифицированной функции гипертангенса в задачах целевой оптимизации

Частным случаем задач оптимального управления является задача минимизации отклонения от заданного целевого значения. При этом могут возникать ситуации, когда в силу дополнительных, сложно формализуемых факторов отклонение в одну сторону не является равнозначным отклонению в другую сторону. Предложенная ФЧПК (8) позволяет формализовать ситуации выбора, когда полезность частного критерия в приближении к заданному целевому значению с разных сторон имеет различающиеся уровни порогов нечувствительности (различную полезность для наихудших значений справа и слева от заданного целевого значения, рис. 5).

Формализация такой ситуации любыми другими способами ведет либо к увеличению числа параметров ФЧПК (возможно, использованию двух различных ФЧПК), либо к увеличению числа частных критериев и, как следствие, – усложнению задачи.

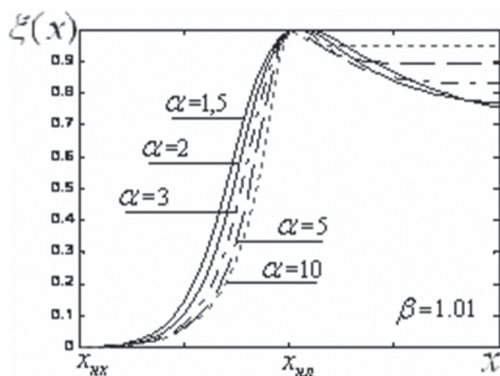


Рис. 5. Функция модифицированного гиперболического тангенса для несимметричной целевой оптимизации (после операции масштабирования)

Выводы

Результаты экспериментальных исследований показали, что все рассмотренные ФПЧК позволяют реализовать требуемые зависимости от значений частных критериев (выпуклую, вогнутую, S-образную) и имеют линейную или квадратичную временную сложность. Наименьшую временную сложность имеет функция Гаусса, несколько большую – логистическая функция и упрощенная модификация функции гипертангенса, среднюю сложность имеют модифицированная функция Гаусса, функция Харрингтона и модифицированная функция гипертангенса, а наибольшую – функция-склейка. При этом предложенная модификация функции Гаусса имеет на 14,2% меньшую среднюю относительную погрешность аппроксимации полезности. Введение степенного параметра в функцию Гаусса позволяет расширить возможности формализации с ее помощью выпуклых зависимостей, однако увеличивает в среднем на 15,98% время реализации процедуры параметрической идентификации.

Предложенная ФПЧК на основе модифицированной функции гипертангенса (8) может быть использована для формализации ситуаций многокритериального выбора, когда полезность частного критерия в приближении к заданному целевому значению с разных сторон имеет различающиеся уровни порогов нечувствительности (различную полезность для наихудших значений справа и слева от заданного целевого значения).

Направлениями дальнейших исследований в этой области могут стать исследования эффективности применения предложенных функций в моделях общей полезности и методах ее идентификации на основе компараторной технологии.

Список литературы: 1. Овезгельдыев, А.О. Синтез и идентификация моделей многофакторного оценивания и оптимизации [Текст] / А.О. Овезгельдыев, Э.Г. Петров,

К.Э. Петров. – К.: Наук. думка, 2002. – 164 с. 2. Камулев, А.Н. Математические методы в системах поддержки принятия решений [Текст] / А.Н. Камулев, Н.А. Северцев. – М.: Высшая школа, 2005. – 311 с. 3. Алтунин, А.Е. Модели и алгоритмы принятия решений в нечетких условиях: Монография [Текст] / А.Е. Алтунин, М.В. Семухин – Тюмень: Издательство Тюменского государственного университета, 2000. – 352 с. 4. Петров, Э.Г. Формирование функций полезности частных критериев в задачах многокритериального оценивания [Текст] / Э.Г. Петров, В.В. Бескоровайный, В.П. Пискалова // Радиоэлектроника и информатика. – 1997. – №1. – С. 71-73. 5. Derringer, G.C. Simultaneous Optimization of Several Response Variables / G.C. Derringer, R. Suich // Journal of Quality Technology. – 1980. – № 12. – P. 214-219. 6. Брахман, Т.Р. Многокритериальность и выбор альтернативы в технике [Текст] / Т.Р. Брахман. – М.: Радио и связь, 1984. – 287 с. 7. Harrington, E.C. The desirability function / E.C. Harrington // Industrial Quality Control. – 1965. – № 21. – P. 494-498. 8. Соболева, Е.В. S-образная функция полезности частных критериев для многофакторной оценки проектных решений [Текст] / Е.В. Соболева // Материалы XIII Международного молодежного форума «Радиоэлектроника и молодежь в XXI веке». – Харьков, 2009. – С. 247. 9. Бескоровайный, В.В. Метод решения общей задачи компараторной идентификации моделей многофакторного оценивания [текст] / В.В. Бескоровайный, Э.Г. Петров, И.В. Трофименко // Бионика интеллекта. – 2006. – № 2 (65). – С. 3-7.

Поступила в редколлегию 05.03.2010 г.

УДК 519.688: 004.896

Идентифікація часткової корисності багатфакторних альтернатив за допомогою S-подібних функцій / В.В. Бескоровайный, Е.В. Соболева // Біоніка інтелекту: наук.-техн. журнал. – 2010. – № 1 (72). – С. 50–54.

У якості функції корисності часткових критеріїв для задач багатокритеріальної оптимізації пропонується модифікація логістичної функції, яка описує сімейство S-подібних залежностей. Досліджується функція корисності часткових критеріїв, започаткована на функції Гауса з додатковим степеневим параметром. Здійснено порівняльний аналіз запропонованих моделей з відомими моделями (у вигляді функцій Гауса, Харрінгтона, логістичної функції, склейки двох степеневих функцій), наведено результати експериментального дослідження точності та складності процедур їх параметричної ідентифікації.

Табл. 1. Іл. 5. Бібліогр.: 9 найм.

UDK 519.688: 004.896

Identification of multiobjective alternative's single utility by S-shaped desirability functions / V. Beskorovainyi, E. Soboleva // Bionics of Intelligence: Sci. Mag. – 2010. – № 1 (72). – P. 50–54.

It's proposed modifications of logistic and Gaussian functions (as S-shaped models of desirability function) for multiobjective optimization. Carried out a comparative analysis of proposed models with known models (Gaussian, Harrington's, logistic and combination of two power functions), are given results of experimental research of accuracy and difficulty of procedures for their parametric identification.

Tabl. 1. Fig. 5. Ref.: 8 items.

УДК 681.3.06



МЕТОДИ ПІДВИЩЕННЯ ЯКОСТІ ЗАХИСТУ ІНФОРМАЦІЇ НА ОСНОВІ ВИКОРИСТАННЯ ФРАКТАЛЬНИХ ФУНКЦІЙ

В.П. Авраменко¹, В.П. Ткаченко², А.Д. Чібіреєв³

^{1, 2, 3} ХНУРЕ, м. Харків, Україна

Сформульовано прямі і зворотні задачі підвищення якості захисту текстової і графічної інформації. Розроблено і удосконалено методи та інструментальні засоби підвищення якості захисту інформації на основі застосування спеціалізованих фрактальних функцій.

ЗВОРОТНІ ЗАДАЧІ ЯКОСТІ, ХАРАКТЕРИСТИКИ ФРАКТАЛІВ, ЯКІСТЬ ЗАХИСТУ ІНФОРМАЦІЇ, АЛГЕБРАІЧНІ І ФРАКТАЛЬНІ ФУНКЦІЇ, РЕГУЛЯРИЗАЦІЯ ОБРОБКИ ДАНИХ

Вступ

В сучасних умовах активізується і одночасно ускладнюється проблема захисту інформації, адже несанкціонований доступ, спотворення, копіювання, знищення інформації наносить шкоду не тільки процесам державного управління, але й інтересам фізичних осіб. Як наслідок, зростає відповідальність за ухвалення прийнятих рішень у ситуаціях, коли окремі похибки здатні призвести до тяжких наслідків у сфері економіки, фінансів, екології та інших сферах.

Проблему підвищення якості систем захисту інформації можна розглядати з точок зору економіки, науки і техніки, які обумовили бурхливий розвиток обчислювальної техніки, інформатики, мікроелектроніки, телекомунікації, тощо. Значний вклад у розвиток методів захисту інформації внесли такі вітчизняні і зарубіжні науковці: І.Д. Горбенко, А.А. Молдован, В.М. Рудницький, В.Ф. Шаньгін, Б. Шнайдер та інші.

В науковому напрямку захисту інформації на теперішній час існує низка проблем, вирішення яких має важливе загальнодержавне і науково-технічне значення. Однією з таких проблем вважається підвищення якості та ефективності захисту друкованої текстової і графічної інформації на основі використання спеціалізованих логічно-алгебраїчних та фрактальних функцій, а також оперативної обробки конфіденційної інформації.

Найважливішим методом захисту друкованої продукції від підробки вважається захист цінних паперів на стадії розробки дизайну і формування зображення із застосуванням тонких графічних елементів – сіток, розеток, віньєток, прихованих елементів, тощо. Такий метод захисту ґрунтується на складності відтворення і репродукції зображень, які пов'язані з геометричною структурою і мінімально можливою товщиною ліній [1].

Візуально порівнюючи підробки з оригіналом, вже при 5–10-кратному збільшенні можна побачити значні викривлення зображення – потовщення ліній, рвані лінії, нечіткість окремих елементів, подвоєння зображення, тощо. Для захисту бланків цінних паперів і документів строгої звітності засто-

совують такі графічні елементи, як спеціальні лінії растру, мікрографіка, реєстрові елементи, приховані елементи, призматичний друк та інші.

1. Постановка задачі захисту інформації

Виконаний авторами аналіз стану проблеми підвищення якості захисту створюваної та виготовленої поліграфічної продукції [1] свідчить про те, що зазначена тема на сьогодні вважається важливою і актуальною, вона дозволяє підвищити якість захисту інформації на основі використання спеціалізованих алгебраїчних і фрактальних функцій.

Мета наукового дослідження полягає у тому, щоб розробити алгоритмічне і програмне забезпечення для створення логіко-алгебраїчних і фрактально-геометричних зображень, котрі зорієнтовані на захист поліграфічної продукції.

Для досягнення поставленої мети сформульовані і вирішуються такі *наукові завдання*:

- визначаються прямі і зворотні завдання моделювання захисту інформації;
- визначаються поняття і основні геометричні характеристики фракталів;
- розробляються системи ітерованих функцій для формування фракталів;
- здійснюється порівняння існуючих і досліджених методів захисту інформації на основі алгебраїчних функцій регулярних подій та фрактальних функцій;
- здійснюється апробація досліджених методів захисту інформації на прикладі фрактального стиснення зображень.

Цінні папери і документи суворого обліку, створені людиною або групою людей, завжди можуть бути повторені іншими. Проте проблема захисту таких документів не безнадійна, коли мати на увазі економічний і часовий чинники. Якщо зробити так, щоб підробка обходилася у декілька разів дорожче за оригінал або виробництво займало б багато часу, то здійснювати таку підробку втратить усякий сенс.

Принципи захисту друкованої продукції базуватися на таких чинниках [1]:

- собівартість захищеного друкованого продукту повинна бути мінімальною, особливо це сто-

сується етикеточної та пакувальної продукції, яка сама по собі йде в навантаження до упакованого продукту і збільшує його собівартість;

– ступінь захисту повинен бути узгодженим з вартістю продукту, що захищається.

Перевищений захист на рівні банківського сертифікату не виправдовує себе для горілчаної етикетки. І навпаки, спрощений захист, застосований для етикетки, знецінить фінансовий документ і підштовхне шахраїв до підробок.

Існуючі натеper принципи захисту поліграфічної продукції поділяються на такі групи:

1. Захист на стадії дизайну за допомогою прийомів верстки і обробки зображень.

2. Технологічні способи друку (орловський, ірисовий друк та інші).

3. Захист за рахунок особливостей паперу, на якому здійснюється друк.

4. Захист за допомогою фарб або інших носіїв друкарської інформації.

5. Використання додаткових фінішних і обробних процедур після друку.

Для захисту цінних паперів на стадії створення оригінал-макету наразі найчастіше застосовують растрову і векторну графіку, а перспективними методами вважаються методи захисту інформації на основі використання алгебраїчних і фрактальних функцій.

2. Визначення та основні характеристики фракталів

Фрактал у вузькому сенсі являє собою складний геометричний об'єкт, який будується в результаті ітераційного циклу, причому окремі частини фрактала подібні за формою усьому фракталу. У широкому сенсі фрактал – це множина точок в евклідовому просторі, що має дробову метричну вимірність, або метричну вимірність, строго більшу від топологічної.

Серед поширених фракталів прийнято виділяти три основні групи [2–4]:

– *Алгебраїчні фрактали*, котрі створюються за допомогою нелінійних обчислювальних процесів в n -вимірних просторах. До цієї групи, зокрема, належить множина Мандельброта.

– *Геометричні фрактали*, котрі у двовимірному випадку створюються за допомогою ламаної-генератора. За один крок алгоритму кожен із відрізків ламаної замінюється на ламану-генератор і таким чином у відповідному масштабі створюється геометричний фрактальний образ. До цієї групи фракталів відносяться тріадна крива Коха і дракон Хартера-Хейтуея.

– *Стохастичні фрактали*, котрі створюються ітераційним процесом з випадковими параметрами. Образи стохастичних фракталів дуже схожі на природні несиметричні дерева.

Фракталам притаманні такі характерні властивості:

– Побудова фрактала здійснюється за допомогою рекурсивної процедури.

– Форма отриманого зображення суттєво залежить від заданих параметрів.

– Самоподібність, тобто подібність частин фракталу до форми усього фракталу.

– Дробово-метрична вимірність або метрична вимірність, що перевершує топологічну.

– Складність фрактального зображення при його збільшенні не змінюється. Саме ця властивість використовується для захисту друкованої текстової і графічної інформації.

– Нетривіальність структури на всіх шкалах: якщо розглянути невеликий фрагмент регулярної фігури в дуже крупному масштабі, то він буде схожий на фрагмент прямої.

– Збільшення масштабу фракталу не веде до спрощення структури, на всіх шкалах можна побачити картину однакової складності.

Багато природних об'єктів володіють фрактальними властивостями – хмари, крона дерев, берегова лінія. Фрактали використовуються також в системах передачі даних для синтезу процедур стиснення і захисту інформації за різними критеріями (фрактальними функціями).

У геометричних фракталів форма описується як послідовність певних геометричних операцій. Наприклад крива Коха стає фракталом в результаті ітерацій, під час яких виконується ділення відрізка прямої на три частини. Далі кожний отриманий фрагмент знову ділиться на три частини і так далі. Поступово лінія стає подібною до сніжинки.

Деякі важливі властивості фракталів відкриваються при дослідженні ітерованих відображень. Дослідження починаються з вибору функції $y = f(x)$ і відшукуються значення послідовності функцій типу $f(x), f(f(x)), f(f(f(x))), \dots$ Прикладом можуть слугувати функції Бесселя для послідовності типу $\sin(x), \sin(\sin(x)), \sin(\sin(\sin(x))) \dots$ Дослідження поведінки коренів комплексної функції на комплексній площині виконані також Артуром Келі.

Термін *фрактал* вперше введено у 1975 році французьким математиком Бенуа Мандельбротом. Ключовою властивістю фрактала вважається самоподібність, котра полягає у тому, що частина структури подібна цілому. Звідси випливає, що фрактал являє собою об'єкт безкінечної складності, котрий дозволяє розглядати стільки ж своїх деталей зблизька, як і здалека.

За фрактал можна прийняти множину Мандельброта, для кожної точки якої треба виконати цикл ітерацій за формулою

$$z_{k+1} = z_k^2 + z_0, k = 0, 1, \dots, n.$$

Величини z_k та z_0 – комплексні числа; $z_k = x_k + iy_k$, причому стартові значення x_0, y_0 – координати точки зображення. Для кожної точки зображення ітерації виконуються обмежену кількість разів (n) або до тих пір, доки модуль комплексного числа z_k не перевищує фіксоване значення 2. Цикл ітерацій для фрактала Мандельброта зазвичай вибирається у діапазоні $x = (-2, 2; +1, 0)$, $y = (-1, 2; +1, 2)$. Для отримання точок зображення у растрі необхідно перерахувати компоненти цього діапазону у піксельні координати.

В загальному випадку обчислювальна процедура перетворення задається таблицею модельних точок, а самі перетворення застосовуються спочатку до заданої базової фігури, а потім до чергової її частини. Таблиця модельних точок містить значення координат (x, y) , де змінюються параметри прямої. Застосовуючи такі дії для побудови фракталів до замкнених кривих або фігур, отримують зображення типу «сніжинка», «зірочка», «листочок», тощо.

Зважаючи на те, що параметри фрактала можна вибирати за допомогою випадкових процесів, а також використовувати у сформованому зображенні кольори та напівтони, різноманітність форм, що задаються за допомогою фракталів, можна вважати нескінченною.

3. Системи ітерированих функцій для створення фракталів

Задачі дослідження операцій прийнято поділяти на прямі та зворотні [4, с. 25–26]. До прямих належать задачі, котрі відповідають на запитання: які кількісні та якісні показники матиме досліджуваній процес або створюваний об'єкт при заданих умовах функціонування?

Пряма задача моделювання фракталів полягає у тому, щоб автоматично виконати поставлену задачу і створити потрібне фрактальне зображення з заданими показниками якості. Для вирішення задачі підвищення якості захисту інформації слід побудувати математичну модель, яка дозволить оцінити показники ефективності (адекватності графічного зображення, вартості та інші) через задані умови і елементи рішення.

Зворотні задачі відповідають на запитання: як вибрати рішення для того, щоб сукупний показник ефективності виконуваної операції приймав найліпше значення? Розв'язувати прямі задачі, зазвичай, простіше за обернені, але це не стосується моделювання фракталів. Для розв'язування зворотної задачі перш за все слід вміти вирішувати пряму задачу.

Зупинимось докладніше на процедурах вирішення зворотних задач. Якщо кількість варіантів рішення невелика, то без зайвих труднощів можна обчислити сукупність локальних критеріїв і на основі їх порівняння знайти один або декілька оптимальних варіантів, для яких глобальна цільо-

ва функція досягає найліпшого значення. Такий спосіб знаходження оптимального рішення називається «простим перебором».

Коли кількість можливих варіантів локальних рішень велика, тоді пошук серед них найліпшого варіанту простим перебором часто практично неможливий. У таких випадках застосовуються методи «направленого перебору», котрі знаходять оптимальне рішення методом послідовних «спроб» або «наближень», з яких кожний подальший крок наближає нас до найліпшого багатокритеріального або багатоальтернативного рішення.

В математиці існує два таких різних об'єкта, як фрактали і хаос, котрі, з однієї сторони, ґрунтуються на різних основах [2, с. 6]: теорія фракталів опирається на геометрію і теорію вимірності, а теорія хаосу – на турбулентні динамічні системи, а з другої сторони, між їх властивостями існує певний взаємозв'язок. Термін «фрактал» відноситься до статичної геометричної конфігурації, такої як миттєвий знімок водоспаду. Термін «хаос» використовується для описання явищ, подібних до турбулентного поведіння погоди.

Дослідники хаосу в минулому столітті виявили [4, с. 33], що хаотичні системи здатні створювати дивовижні зображення (вирішують прямі задачі моделювання фракталів), які були названі дивними аттракторами. Зокрема Майкл Барнслі першим висунув ідею (зворотну задачу), що можна генерувати наперед задане зображення як аттрактор хаотичного процесу.

Сутність зворотної задачі моделювання фракталів полягає у тому, щоб для заданого зображення побудувати хаотичну систему функціональних перетворювачів, для якої задане зображення буде дивними аттрактором. Запропоновану спеціальну систему відображень Барнслі назвав *системою ітерированих функцій* (Iterated Function System, IFS).

За допомогою IFS можна створювати реалістичні зображення природних об'єктів, таких як листя папороті, дерева; при цьому багаторазово використовуються перетворення, які переміщують, змінюють в розмірі і повертають частини зображень. В IFS використовується самоподібність, а фрактальні об'єкти моделюються як композиція сукупності найдрібніших копій самого себе. IFS-фрактали використовуються також для стиснення зображень.

4. Фрактальне стиснення текстових і графічних зображень

Фракталами з часовим порогом можна моделювати поведінку хаотичних динамічних розподілених систем (наприклад, стік річок), в яких невеликі зміни вхідних дій (наприклад, підвищення довколишньої температури або теплий дощ при розтаненні снігу) можуть обумовити великі зміни вихідних параметрів (наприклад, бурхлива повідь).

IFS-фрактали використовуються для стиснення зображень, а фрактальний метод дає кращі результати при багатократному стисненні, ніж метод JPEG та інші методи стиснення, з малими збитком якості зображення. Слід відзначити, що проблема стиснення зображень не вирішується повністю фрактальним стисненням, для цього також використовується й інші сучасні підходи.

Революція у стисненні зображень відбулася у зв'язку з виходом в 1977 році книги Мандельброта «Фрактальна геометрія природи». Процес фрактального стиснення зображень на основі системи ітерированих функцій передбачає виконання таких етапів [6, с. 217]:

1. Розділити ціле зображення на домени таким чином, щоб складові частини зображення не перекривалися. Набір доменів повинен перекривати все зображення повністю.

2. Задати набір рангових областей зображення. Рангові області можуть перекриватися. Вони не повинні обов'язково закривати всю поверхню зображення.

3. Виконати фрактальні перетворення. Для кожного домена підібрати таку рангову область, яка після афінного перетворення найточніше апроксимує домен. Таке перетворення не тільки стискає і деформує зображення рангової області, але й змінює яскравість і контраст.

4. Виконати стиснення і зберегти параметри афінного перетворення. Файл із стислим зображенням має дві частини: заголовок з інформацією про розташування доменів і рангових областей і ефективно упаковану таблицю коефіцієнтів афінних перетворень для кожного домена.

Етапи відновлення зображення такі [6, с. 218]:

1. Створити два зображення А і В однакового розміру, котрі можуть не бути рівними розміру початкового зображення. Початковий рисунок областей А і В не має значення.

2. Перетворити дані і перенести їх з області А в область В. Для цього слід спочатку розділити зображення В на домени так само, як це було зроблено на першій стадії процесу стиснення (розташування доменів описано в заголовку файлу). Для кожного домена області В виконати відповідне афінне перетворення рангових областей зображення А, яке описано коефіцієнтами із стислого файлу, і результат помістити в область В.

3. Перетворити дані з області В в область А. Цей крок ідентичний попередньому, тільки зображення В і А помінялися місцями, тобто тепер розділяється на блоки область А і на ці блоки відображаються рангові області зображення В.

4. Багаторазово повторити другий і третій кроки процедури відновлення даних до тих пір, поки зображення А і В не стануть нерозрізненими. Точність відповідності залежить від точності афінного

перетворення, коефіцієнти якого обчислюються в процесі стиснення.

Алгоритми стиснення і відновлення інформації використовують цілочислову арифметику і методи зменшення накопичуваних помилок округлення. На відміну від поширених в даний час методів стиснення-відновлення графічних зображень, фрактальне перетворення асиметричне: процес відновлення не можна провести шляхом простої інверсії процедури стиснення. Стиснення вимагає набагато більшої кількості обчислень, ніж відновлення.

В результаті фрактального перетворення створюється набір цифр, який в дуже стислій формі описує зображення. Великий коефіцієнт стиснення прискорює передачу високоякісних зображень. При високому рівні розв'язувальної можливості початкового зображення фрактальне відображення набагато ефективніше з погляду зниження об'єму стислої інформації.

Зазначеним чином здійснюється фрактальне стиснення зображень для прискореної передачі і економного збереження інформації. Цей метод [7] здійснює послідовне ітеративне обчислення координат нових точок простору у відповідності до заданих виразів:

$$x_{k+1} = F_x(x_k, y_k),$$

$$y_{k+1} = F_y(x_k, y_k),$$

де F_x, F_y – функції перетворення координат, наприклад афінного.

Ці функції безпосередньо визначають кінцеву форму фрактала. Наприклад зображення рослин можна сформувати послідовними геометричними перетвореннями зміщення, повороту та масштабу відрізків прямих з відповідними параметрами [8]. Метод IFS застосовується не тільки для створення зображень. Його використовують також для ефективного стиску зображень під час запису у файл.

5. Текстова регуляризація в системах обробки даних

Важливою вимогою до текстових перекладачів і трансляторів є адекватність перекладів, яка визначається семантичною повнотою, точністю і стилістичною ефективністю, що включає принцип відповідності тексту перекладу стилістичним нормам мови перекладу. За цими критеріями оцінюється якість перекладу. На сьогодні проблема оцінювання якості перекладу ще не формалізована. Практичні оцінки якості перекладу визначаються за співвідношеннями, що базуються на особистому професійному досвіді редактора або перекладача.

Для підвищення якості процедур обробки даних знаходить застосування текстова регуляризація, яка може розповсюджуватися на декілька рівнів. Існує трибичний взаємозв'язок між регулярними виразами, кінцевими автоматами і регулярними

граматиками мови. Описання мови часто базується на використанні регулярних виразів, на основі яких можна побудувати кінцеві автомати типу лексичних аналізаторів мови. Такі лексичні аналізатори перетворюють послідовності символів початкового файлу у сукупність елементів, типи яких визначаються видом регулярних виразів.

Нижче наведено приклад програми на flex для розбиття тексту російської мови на слова, абрєвіатури, числа, класи знаків пунктуації [9]:

```
digit      [0-9]
intconst  [+\\-]?{digit}+
realconst [+\\-]?{digit}+\\. {digit}+(e[+\\-]?{digit}+)?
letter    [a-я]
capital  [А-Я]
word      ({letter}|{capital}){letter}*{letter}+{letter}+
abbreviation {capital}*
punctuation {,|:|;|-|(|)}{ }| []
stops     {.!|?}
%%
{intconst}  Процедура формування лексеми – ціле число;
{realconst}  Процедура формування лексеми – дробове число;
{word}      Процедура пошуку слова в словнику і формування лексеми слово;
{abbreviation}  Процедура формування лексеми абрєвіатура;
{punctuation}  Процедура формування лексеми – розділовий знак;
{stops}     Процедура формування лексеми – ознака кінця пропозиції;
%%
```

Як інструменти обробки даних можуть використовуватися різні формальні подання, котрі задовольняють обмеженням, що накладаються на дане подання. Нижче наведено приклад регулярного подання мовної конструкції «Програмний продукт буде графік функції» на основі семантичних функцій:

$$f_1(V_3^2(y), V_1^1(x), f_3^4(V_1^4(z), V_1^4(u))),$$

де f_i – функції, що визначають відносини в пропозиції; $V_j(x)$ – функція, що визначає j значення аргументу x .

Важливим питанням процедури регуляризації є еквівалентність формального подання початкової мовної конструкції вихідним поданням. Нехай початкова мовна конструкція A перетвориться процедурою регуляризації R у формальне подання B :

$$A \xrightarrow{R} B.$$

Припустимо, що існує формальна процедура R' , що дозволяє одержувати з формального уявлення мовну конструкцію:

$$B \xrightarrow{R'} A'.$$

Важливим при цьому є питання про еквівалентність мовних конструкцій A і A' . Відповідь на це питання залежить від симетричності перетворень R і R' . Симетричність перетворень можуть забезпечити еквівалентні структури аналізу і синтезу на рівні синтаксичного і семантичного аналізу. Головна проблема полягає в адекватності віддзеркалення семантико-синтаксичних зв'язків пропозиції на формальному рівні, їх використання на етапах синтезу і аналізу обробки даних.

Висновки

Проведено дослідження методів, математичних моделей та обчислювальних процедур підвищення якості захисту інформації на основі фрактального підходу. Виявлено, що для створення фракталів необхідно сформулювати обернену задачу моделювання фракталів.

Для вирішення поставленої зворотної задачі запропоновано такий спосіб: для заданого зображення або його фрагментів слід підібрати відповідні коефіцієнти афінного перетворення. Цей метод доцільно використовувати для запису кольорових фотографій у файли зі стиском у сотні разів меншим від початкового без суттєвого погіршення зображення.

Оскільки фрактали представляють дуже складні зображення, а описати їх можна за простими ітеративними формулами, то для збереження опису цих формул потрібний значно менший обсяг інформації, ніж для збереження відповідних растрових зображень.

Наукова новизна виконаних досліджень полягає у тому, що досліджено проблему підвищення якості захисту інформації на основі використання алгебри регулярних подій, фрактальних функцій і мовних регулярних виразів. Потім відбувається перетворення регулярних виразів в кінцеві автомати, котрі реалізують необхідні ітераційні цикли.

Практична цінність виконаних досліджень полягає в доведенні отриманих наукових результатів до конкретних інженерних методик, математичних моделей та інструментальних засобів захисту інформації за рахунок застосування алгебраїчних та фрактальних функцій.

Список літератури: 1. Киричок, П. О. Методи захисту цінних паперів та документів суворого обліку [Текст] / П. О. Киричок, Ю. М. Коростіль, А. В. Шевчук – К.: КПІ, 2008. – 368 с. 2. Кроновер, Р. Фракталы и хаос в динамических системах [Текст] / Р. Кроновер. – М.: Техносфера, 2006. – 488 с. 3. Фрактальный анализ процессов, структур и сигналов [Текст] / Под ред. Р. Э. Пашенко. – Харьков: ЭкоПерспектива, 2006. – 348 с. 4. Уэлстид, С. Фракталы и вейвлеты для сжатия изображений в действии [Текст] / С. Уэлстид. – М.: Триумф, 2003. – 320 с. 5. Вентцель, Е. С. Исследование операций. Задачи, принципы, методология [Текст] / Е. С. Вентцель, – М.: Высшая школа, 2001. – 208 с. 6. Сиденко, Л. А. Компьютерная графика и геометричес-

кое программирование [Текст]: учебное пособие / Л.А. Сиденко. – СПб.: Питер, 2009. – 224 с. 7. *Путятін, Є.П.* Методи та алгоритми комп'ютерного зору: навчальний посібник [Текст] / Є.П. Путятін, В.О. Гороховатський, О.О. Матат. – Харків: Компанія СМІТ, 2006. – 236 с. 8. *Блинова, Т.А.* Компьютерная графика [Текст]: учебное пособие / Т.А. Блинова, В.Н. Порев. – К.: Юниор, СПб.: Корона принт, 2006. – 432 с. 9. *Авраменко, В.П.* Регуляризация процедур обработки данных в системах искусственного интеллекта [Текст] / В.П. Авраменко, Н.А. Валенда // Бионика интеллекта. – 2007. – № 2 (67). – С. 47–50.

Надійшла до редколегії 11.03.2010 р.

УДК 681.3.06

Методы повышения качества защиты информации на основе применения фрактальных функций / В.П. Авраменко, В.П. Ткаченко, А.Д. Чибірев // Бионика интеллекта: научн.-техн. журнал. – 2010. – № 1 (72). – С. 55–60.

Сформулирована проблема повышения качества защиты печатной текстовой и графической информации

на основе применения алгебраических и фрактальных функций. Поставлена обратная задача моделирования фрактальных функций. Разработаны методы и вычислительные процедуры решения обратных задач моделирования фрактальных функций на основе применения аффинных преобразований.

Библиогр.: 9 назв.

UDC 681.3.06

Extended quality security information Methods with operation of fractal functions / V.P. Avramenko, V.P. Tkachenko, a.d. Chibipev // Bionics of Intelligence: Sci. Mag. – 2010. – № 1 (72). – P. 55–60.

The problem of improvement quality protection of printing text and graphic information on the basis of application algebraic and fractal functions is formulated. The reverse problem modelling fractal functions is set. Methods and computing procedures decision of reverse problems modelling fractal functions on the basis of application affine transformations are developed.

Ref.: 9 items.

УДК 518.81



МОДЕЛЬ ЗАДАЧИ РАСПРЕДЕЛЕНИЯ РЕСУРСОВ В УСЛОВИЯХ МНОГОКРИТЕРИАЛЬНОСТИ И НЕОПРЕДЕЛЕННОСТИ

В.В. Крючковский¹, О.А. Пискалова², Д.И. Филипская³

¹ ХНТУ, м. Херсон, Украина, meo@kstu.edu.ua;

² ХНУРЭ, м. Харьков, Украина, st@kture.kharkov.ua;

³ ХНУРЭ, м. Харьков, Украина, st@kture.kharkov.ua

Проведен анализ особенностей решения задач многокритериального принятия решений в условиях неопределенности. Предложена универсальная модель выбора компромиссного решения в условиях многокритериальности и различных типов интервальной неопределенности при решении задач распределительного типа.

РАСПРЕДЕЛЕНИЕ РЕСУРСОВ, МНОГОКРИТЕРИАЛЬНОСТЬ, ИНТЕРВАЛЬНАЯ НЕОПРЕДЕЛЕННОСТЬ, ОПТИМИЗАЦИЯ, МОДЕЛЬ МАРКОВИЦА, VAR И VAL МОДЕЛИ

Введение

Принятие решений является обязательным и неотъемлемым этапом любой целенаправленной деятельности. При этом многие задачи принятия решений могут быть содержательно интерпретированы как задачи распределения ограниченных ресурсов.

В общей постановке задача может быть сформулирована следующим образом.

Пусть субъект S располагает некоторым мноресурсом в количестве D . Имеется некоторое ограниченное множество объектов $O_i, i = \overline{1, n}$, которые претендуют на получение ресурса d_i . При этом спрос превышает предложение:

$$\sum_{i=1}^n d_i \geq D. \quad (1)$$

Получая ресурс d_i , каждый объект генерируют некоторый эффект V_i

$$V_i = F_i(d_i), \quad (2)$$

который возвращается полностью или частично субъекту S . Цель субъекта заключается в принятии такого решения, в результате которого

$$V_S = \max_{d_i \in D} \sum_{i=1}^n V_i(d_i), \quad (3)$$

то есть максимизируется суммарный эффект субъекта.

К такой постановке сводятся задачи формирования инвестиционных портфелей, определения номенклатуры выпускаемой продукции, развития производства и так далее. Для их решения разработаны вычислительные методы, проблемно ориентированные на особенности задач. Определяющим признаком является вид оператора F_i (его часто называют производственной функцией) ограничений, а также счетность или несчетность множества допустимых решений. В этой связи можно указать методы непрерывного и дискретного линейного и динамического программирования, метод ветвей и границ, приближенные численные методы и так далее. Однако при этом, часто по умолчанию,

предполагается, что получаемый эффект V_i является скалярной величиной, а все исходные данные — детерминированы. Такие допущения не всегда корректны и могут приводить к идеализированным, иногда практически бесполезным результатам.

Целью нашей работы является анализ особенностей и обоснование модели решения задач по распределению ресурсов в условиях многокритериальности и неопределенности исходных данных.

Анализ особенностей решения многокритериальных задач принятия решений

Будем полагать, что в общем случае эффекты V_i объектов характеризуются кортежем разнородных, измеренных в различных шкалах, с различным направлением доминирования локальных характеристик (частных критериев)

$$V_i = \langle V_{ij} \rangle, j = \overline{1, m}. \quad (4)$$

Например эффект может характеризоваться экономической, социальной и экологической составляющими. Это означает, что задача (3), с учетом многомерности и противоречивости частных показателей эффективности (4), не имеет единственного решения, то есть является некорректной по Адамару [1].

Чтобы получить единственное решение задачи (3) в многокритериальной постановке, необходимо ее регуляризовать, то есть каким-либо образом привести к скалярному виду. В настоящее время существует множество проблемно — ориентированных методов регуляризации, например принцип главного критерия, последовательной оптимизации, анализа иерархий, функционально-стоимостной анализ и так далее [2]. Вместе с этим все перечисленные схемы являются частными случаями подхода, основанного на теории полезности [3]. Эта теория основана на гипотезе, что на множестве частных характеристик решения существует обобщенная многофакторная скалярная оценка вида

$$\bar{V}_i(d_i) = Q(A, V_{ij}(d_i)), \quad (5)$$

где A — кортеж, коэффициентов, приводящих разнородные V_{ij} к изоморфному виду.

Для конструктивной реализации модели (5) необходима ее структурно-параметрическая идентификация. Учитывая, что процедура многофакторного скалярного оценивания является субъективной интеллектуальной процедурой, основным источником (носителем) исходной информации является лицо, принимающее решение, или эксперты, а методом получения этой информации — метод экспертного оценивания. В силу разброса субъективных мнений экспертов, особенно при формировании количественных оценок весовых коэффициентов A , возникает интервальная неопределенность.

Другими источниками неопределенности модели (5) являются так называемые НЕ-факторы [4]: неполнота знаний, невозможность корректной формализации, точного количественного измерения влияющих факторов и так далее. В результате модель (5) принимает вид:

$$\tilde{V}_i(d_i) = Q[\tilde{A}, \tilde{V}_{ij}(d_i)], \quad (6)$$

где знаком « \sim » обозначены неопределенные данные.

Для дальнейшего конструктивного анализа необходимо определить количественные и качественные характеристики неопределенности.

Количественно неопределенности будем характеризовать интервалом возможных значений, то есть областью на числовой оси ограниченной значениями слева и справа. Качественно неопределенности будем классифицировать на основе информации о характере распределения возможных значений на интервале. Наиболее распространенной формой описания неопределенности является статистическое представление, когда распределение величин на интервале задается функцией плотности распределения вероятности и соответствующими статистическими параметрами. Однако во многих случаях вероятностная интерпретация неопределенности оказывается некорректной. Это привело к созданию и бурному развитию теории нечетких множеств, когда возможное распределение значений на интервале задается субъективной функцией принадлежности нечеткому множеству, носителем которого является интервал возможных значений [5]. В отличие от статистической неопределенности, описание которой базируется на объективной информации, функция принадлежности формируется на основе субъективных представлений экспертов и поэтому менее информативна.

Наконец, наименее информативным является представление неопределенности в виде интервальных величин, когда информация о предпочтительности значений на интервале вообще отсутствует [6].

Независимо от вида конкретных неопределенностей принципиальным является обстоятельство, что модель (6), а, следовательно, и модель (3) дают решения в интервальном виде. Это обуславливает необходимость решения дополнительной задачи

по определению точечного решения. В настоящее время существует два подхода к ее решению. Первый основан на детерминизации неопределенностей на этапе постановки задачи. В этом случае на этапе подготовки задачи проводится анализ и детерминизация исходных интервальных неопределенностей путем замены их средними, модальными, централизованными и так далее точечными значениями. Далее задача решается классическими методами в детерминированной постановке. При этом теряется очень ценная информация об интервале возможных решений. Альтернативой является подход, основанный на интервальной анализе.

Современный подход к решению задач оптимального распределения ресурсов в условиях интервальной неопределенности в общем случае основан на двухкритериальном рассмотрении, когда одним из критериев является уровень суммарной эффективности использования ресурсов по всей совокупности объектов, между которыми распределяются ресурсы, а вторым критерием — мера неопределенности (риска) эффективного использования ресурсов. При этом первый критерий подлежит максимизации, а второй — минимизации. Впервые такую математическую двухкритериальную модель задачи оптимального распределения ресурсов в условиях интервальной неопределенности предложил Гарри Марковиц [7], который за цикл работ по портфельному инвестированию получил в 1990 г. Нобелевскую премию. В рамках модели Марковица в качестве критерия уровня суммарной эффективности использования ресурсов (в интерпретации Марковица роль ресурса играет капитал) берется математическое ожидание суммарной эффективности как случайной величины, а в качестве критерия меры неопределенности — дисперсия суммарной эффективности. Такой выбор математического выражения меры неопределенности позволил реализовать в рамках модели Марковица распределение ресурсов по нескольким объектам (диверсификация ресурсов), что при выполнении некоторых условий должно приводить к уменьшению риска. Математическая модель Марковица задачи оптимального распределения ресурсов в условиях неопределенности принадлежит к классу задач квадратичного программирования. Следует подчеркнуть, что класс задач оптимизации, известный как задачи квадратичного программирования, был сформулирован, и развита теория решений такого класса задач в основном под влиянием модели Марковица.

Как известно [8], классическая постановка Марковица задачи формирования оптимальных инвестиционных портфелей является двухкритериальной, один из критериев которой — среднее ожидаемое значение эффективности $m_p \rightarrow \max$, а второй — волатильность (изменчивость) эффективности $\sigma_p^2 \rightarrow \min$.

В последние годы интенсивно развивается более общий подход к решению задач оптимального распределения ресурсов в условиях статистической неопределенности, основанный на расчете вероятности p^* события, состоящего в том, что суммарная эффективность использования ресурсов, трактуемая как случайная величина, примет значение меньше, чем заданный уровень R^* .

Подход, основанный на рассмотрении задач оптимального распределения ресурсов в такой постановке, получил название VaR — подхода (VaR — аббревиатура словосочетания Value-at-Risk). При постановке задач оптимального распределения ресурсов, основанного на VaR-подходе, критерием уровня суммарной эффективности является V_i , а критерием неопределенности (риска) вероятность p^* . По существу концепция VaR соответствует пониманию риска традиционно используемого в технических областях, где величина риска обычно измеряется величиной вероятности наступления неблагоприятной ситуации (вероятность катастрофы, аварии, выхода из строя аппаратуры) или просто неблагоприятного сочетания значений случайных переменных.

Следует подчеркнуть, что VaR подход ориентирован на анализ и принятие решений в условиях вероятностной неопределенности. Вероятностная интерпретация неопределенности во многих случаях некорректна. Поэтому дальнейшее развитие теории принятия решений в условиях неопределенности требует обобщения VaR подхода на случай нечеткой (заданной в виде нечетких множеств) и интервальной (заданной интервальными величинами) неопределенностей.

Во всех перечисленных случаях оценка эффективности решения является интервальной, то есть ограниченной слева и справа, величиной. Задача в этом случае заключается в выборе конкретного точечного решения на интервале решений.

Любому точечному решению соответствует некоторое ожидаемое значение эффекта (Value), которое определяется конкретными точечными значениями переменных. По определению, переменные являются интервальными, то есть могут принимать с некоторой возможностью любые значения на интервале. Отклонение переменных от принятых точечных значений приводит к потерям (Loss). При этом потери могут быть двух видов:

- негативными (L_N), что означает уменьшение эффективности по сравнению с расчетным уровнем за счет неблагоприятного сочетания значений параметров интервальных возможных значений (это аналог традиционного вероятностного риска R);

- позитивными (L_p) — это недополученный эффект, который потенциально можно было бы получить, если бы параметры приняли значения более благоприятные по сравнению с расчетными.

Технологию принятия решений с учетом указанных возможных потерь будем обозначать аб-

бревиатурой (Value-at-Loss) и называть VaL технологией.

Согласно VaL технологии ожидаемый эффект V необходимо максимизировать, а потери обоих видов L_N, L_p — минимизировать. При этом сумма L_N и L_p является постоянной величиной. Таким образом

$$\begin{aligned} V(x) &\rightarrow \max_{x \in X}; \\ L_N(x) &\rightarrow \min_{x \in X}; \\ L_p(x) &\rightarrow \min_{x \in X}; \\ L_N(x) + L_p(x) &= const; \\ a &\leq x \leq b, \end{aligned} \tag{7}$$

где a, b — соответственно левая и правая границы интервала возможных значений переменных x .

Очевидно, что позитивные (L_p) и негативные (L_N) потери имеют для пользователя (ЛПР) различную ценность: L_N — это прямые потери эффекта (финансов, времени выполнения работы, материальных ресурсов и так далее), тогда как (L_p) — это сожаление о недополученном потенциально возможном эффекте. Тогда с учетом того, что

$$L_N(x) + L_p(x) = const, \tag{8}$$

можно записать

$$x^\circ = \arg \max_{x \in X} [V(x) - \alpha L_p(x) - (1 - \alpha)L_N(x)] \tag{9}$$

при $0 \leq \alpha \leq 1$.

Тогда, если $\alpha = 0$, реализуется стратегия крайнего пессимизма, при $\alpha = 1$ — стратегия крайнего оптимизма, при $\alpha = 0,5$ — минимаксная стратегия.

Для вычисления L_N и L_p будем полагать, что известна зависимость, характеризующая распределение на интервале возможных значений переменных $x \in [a, b]$.

$$\begin{aligned} V &= F(x), \\ x &\in [a, b]. \end{aligned} \tag{10}$$

Тогда при $x = a$:

$$\begin{aligned} L_p(x) &= 0; \\ L_N &= \int_a^b F(x) dx; \end{aligned} \tag{11}$$

при $x = b$:

$$\begin{aligned} L_N(x) &= 0; \\ L_p &= \int_a^b F(x) dx; \end{aligned} \tag{12}$$

При $x = c, a \leq c \leq b$:

$$\begin{aligned} L_p &= \int_a^c F(x) dx; \\ L_N &= \int_c^b F(x) dx. \end{aligned} \tag{13}$$

В качестве функции (10) могут выступать не только традиционные для VaR технологии плотности распределения вероятностей, но и функции принадлежности неопределенностей заданных в виде нечетких множеств. При этом интервальные величины могут быть корректно интерпретированы или как нечеткие множества с толерантной на интервале неопределенности функцией принадлежности равной $\mu(x) = 1$, или как статистические неопределенности с равновероятным законом распределения значений на интервале. Такой подход делает VaL технологию универсальной и методологически корректной для любых видов интервальной неопределенности: статистической, нечеткой или представленной в виде нечетких величин.

Выводы

В связи с широким распространением вычислительной техники и успехами теории математического программирования в области создания проблемно-ориентированных методов оптимизации в настоящее время пользователи не сталкиваются с принципиальными вычислительными трудностями. Поэтому усилия разработчиков сосредоточены в основном на методологических аспектах синтеза, повышении точности и адекватности исходных математических моделей за счет более полного и глубокого учета влияющих факторов, их взаимосвязи, нелинейности, нестационарности и так далее. Учет перечисленных факторов приводит к необходимости учета многокритериальности целевых функций и неопределенности исходной информации.

Следует подчеркнуть, что в настоящее время в большинстве случаев аспекты многокритериальности и неопределенности рассматриваются вне их взаимосвязи. Методы принятия решения в условиях риска, например широко известные VaR технологии, по умолчанию предполагают, что эффект является скалярной величиной. В свою очередь при решении задач многокритериальной оптимизации в большинстве случаев рассматриваются детерминированные постановки. Еще одна особенность современных подходов заключается в том, что в качестве неопределенности в основном рассматриваются статистические виды неопределенности и соответствующие им риски, что делает невозможным корректный учет других видов неопределенности, таких как нечеткая и интервальная.

В настоящей работе рассмотрен подход, ориентированный на комплексный учет многокритериальности и различных видов неопределенности при синтезе универсальной модели решения задач распределительного типа. Предложено обобщение и развитие известной VaR технологии принятия решений в условиях статистической неопределенности на случай других типов неопределенности.

Предложенная VaL технология отличается тем, что точечное решение из интервала возможных значений определяется (с учетом трех частных критериев: собственно эффективности, возможных позитивных и негативных потерь) как многокритериальная оптимизационная задача. Дальнейшие исследования должны быть направлены на определение методов взаимной корректной трансформации интервальной неопределенностей различного типа с целью вычисления эквивалентной интервальной неопределенности для исходных данных смешанного типа.

Список литературы: 1. *Математический энциклопедический словарь* [Текст] / под ред. Ю.В. Прохорова. — М.: Сов. энциклопедия, 1988. — 250 с. 2. *Петров, Е.Г.* Методи і засоби прийняття рішень в соціально-економічних системах [Текст] / Е.Г.Петров, М.В.Новожилова, І.В.Гребенік. — К.: Техніка, 2004. — 256 с. 3. *Фишберн, П.* Теория полезности для принятия решений [Текст] / П. Фишберн. — М.: Наука, 1978. — 352 с. 4. *Нариньяни, А.С.* НЕ-факторы: неоднозначность (доформальное исследование) [Текст] / А.С. Нариньяни // Новости искусственного интеллекта. — 2003. — №5. с. 58-69. 5. *Заде, Л.* Понятие лингвистической переменной и его применение к принятию приближенных решений [Текст] / Л. Заде. — М.: «Мир», 1976. 6. *Алефельд, Г.* Введение в интервальные вычисления: Пер. с англ. [Текст] / Г. Алефельд, Ю. Херцбергер. — М. Мир, 1987. — 360 с., ил. 7. *Шарп, У.Ф.* Инвестиции [Текст] / У.Ф. Шарп, Г.Дж. Александер, Д.В. Бэйли. — М.: Инфра-М, 2001. 8. *Крянев, А.В.* Основы финансового анализа и портфельного инвестирования в рыночной экономике [Текст] / А.В. Крянев. — М.: МИФИ, 2001.

Поступила в редколлегию 16.03.2010 г.

УДК 518.81

Модель задачі розподілу ресурсів в умовах багатокритериальності й невизначеності / В.В. Крючковський, О.О. Писклакова, Д.І. Філіпська // Біоніка інтелекту: наук.-техн. журнал. — 2010. — № 1 (72). — С. 61–64.

У статті обґрунтовано модель рішення задач з розподілу ресурсів в умовах багатокритериальності й невизначеності вхідних даних. Стверджується, що у запропонованій VaL технології точкове рішення визначається з урахуванням трьох часткових критеріїв, тобто як у багатокритериальній оптимізаційній задачі.

Бібліогр.: 8 найм.

UDK 518.81

Model of the allocation problem of resources under the conditions of multicriteriaity and uncertainty / V.V. Kruchkovsky, O.A. Pisklakova, D.I. Filipkaya // Bionics of Intelligence: Sci. Mag. — 2010. — № 1 (72). — P. 61–64.

In the article the model of solution of problems by the distribution of resources under the conditions of multicriteriaity and uncertainty of initial data is substantiated. It is asserted that in that proposed of VaL of technology the point solution it is determined taking into account three particular criteria, i.e., as in multicriterional optimization task.

Ref.: 8 items.

УДК 519.673



ПОИСК ИЗМЕНЕНИЯ СЦЕН В ВИДЕОДАНЫХ НА БАЗЕ АНАЛИЗА РЕЗУЛЬТАТОВ СЕГМЕНТАЦИИ

С.В. Машталир¹, С.В. Постульга², К.С. Щербинин³^{1, 2, 3} ХНУРЭ, г. Харьков, Украина, mashtalir_s@kture.kharkov.ua

Рассмотрена задача обнаружения изменения характеристик временных рядов на примере видеоданных при помощи анализа главных компонент, проводившегося на результатах сегментации исходных видеоданных алгоритмом JSeg. Анализ главных компонент проводился с помощью преобразования Карунена-Лоева. Также рассмотрено влияние параметров JSeg на отслеживание изменений свойств рядов.

МНОГОМЕРНЫЕ ВРЕМЕННЫЕ РЯДЫ, ПРЕОБРАЗОВАНИЕ КАРУНЕНА-ЛОЕВА, ВИДЕОДАНЫЕ

Введение

Обнаружение изменения свойств временных рядов и их сегментации достаточно давно привлекает внимание исследователей [1–5], а для ее решения к настоящему времени сложился ряд относительно независимых подходов, среди которых, в первую очередь, необходимо отметить статистический (корреляционный, спектральный, регрессионный анализы), адаптивный (экспоненциальное сглаживание, анализ следящего сигнала) и основанный на методах классификации (кластеризация, сегментация). И если для одномерных рядов разработаны достаточно эффективные алгоритмы анализа [1–4], результаты, относящиеся к многомерным последовательностям, которыми в целом и являются видеоданные, значительно скромнее [5]. Вместе с тем, в важных приложениях, связанных с контролем производственных процессов, биоинформатикой, распознаванием образов, медицинской диагностикой, телеметрией и, особенно, с обработкой видео сигналов приходится иметь дело с многомерными сигналами и необходимостью их анализа. В связи с этим можно говорить о такой важнейшей задаче, как разбиение временных рядов на относительно однородные сегменты и обнаружение момента перехода от одного сегмента к другому.

Поскольку многомерные временные ряды, а особенно видеоданные, как правило содержат избыточную информацию, их анализ может осуществляться на основе контроля некоторых обобщенных характеристик, таких, как спектры, корреляционные функции, главные компоненты и тому подобное.

Задача радикально усложняется в случае, когда объем выборки N не фиксирован, количество сегментов s априори неизвестно, а сами данные $x(k)$ поступают на обработку последовательно в реальном времени. Именно такая ситуация наиболее характерна в задачах мониторинга реально функционирующих объектов управления, оперативной медицинской диагностики, обработки видео сигналов. Здесь на первый план выступает уже

не собственно сегментация, а последовательное обнаружение моментов изменения свойств многомерного сигнала. Таким образом, мы приходим к тому, что существует необходимость анализа видеоданных (многомерных временных рядов) с целью выявления изменений сцены, то есть изменения свойств. Одним из подходов, позволяющих решить данную задачу, является подход, основанный на переходе к одномерным временным рядам при помощи анализа главных компонент. Следовательно, **целью данной работы** является разработка и реализация метода обнаружения изменений свойств многомерных временных рядов при помощи нейронных сетей, что может быть использовано для поиска моментов изменения сцен и обнаружения ключевых кадров, а именно метода главных компонент. Следует отметить, что в качестве начальной информации для анализа мы берем результаты сегментации исходных видеоданных.

1. Обнаружение изменения свойств многомерных временных рядов на основе анализа главных компонент

Важной проблемой при анализе больших массивов (как по объему, так и по размерности) наблюдений, заданных в форме временных рядов, является задача их сжатия с целью выделения латентных факторов, определяющих внутреннюю структуру контролируемого сигнала, что в конечном итоге преследует цель сделать исходный временной ряд более просто интерпретируемым с точки зрения обнаружения изменений свойств.

Одним из наиболее эффективных подходов к решению этой задачи является аппарат факторного анализа [6], в рамках которого широкое распространение получил метод главных компонент, особенно в задачах распознавания образов, обработки изображений, спектрального анализа и тому подобного и известного еще как преобразование Карунена-Лоева.

Исходной информацией для анализа является $k \times n$ матрица наблюдений

$$X = \begin{pmatrix} x_1(1) & x_2(1) & \dots & x_n(1) \\ x_1(2) & x_2(2) & \dots & x_n(2) \\ \vdots & \vdots & \ddots & \vdots \\ x_1(u) & x_2(u) & \dots & x_n(u) \\ \vdots & \vdots & \ddots & \vdots \\ x_1(k) & x_2(k) & \dots & x_n(k) \end{pmatrix}, \quad (1)$$

образованная массивом из k n -мерных векторов наблюдений $x(u) = (x_1(u), x_2(u), \dots, x_n(u))^T$, и ее корреляционная ($n \times n$) матрица вида

$$R(k) = \frac{1}{k} \sum_{u=1}^k (x(u) - \bar{x})(x(u) - \bar{x})^T = \frac{1}{k} \sum_{u=1}^k x^C(u) x^{CT}(u),$$

где $x^C(u) = x(u) - \bar{x}$ центрированные относительно среднего исходные данные.

Метод главных компонент состоит в проецировании наблюдаемых исходных данных из исходного n -мерного пространства в m -мерное ($n > m \geq 1$) выходное и сводится к нахождению системы w^1, w^2, \dots, w^m ортогональных собственных векторов матрицы $R(k)$ таких, что $w^1 = (w_1^1, w_2^1, \dots, w_n^1)^T$ соответствует наибольшему собственному значению λ матрицы $R(k)$, $w^2 = (w_1^2, w_2^2, \dots, w_n^2)^T$ – второму по величине собственному значению λ и так далее. Иначе говоря, речь идет об отыскании решения матричного уравнения

$$(R(k) - \lambda_l I) w^l = 0$$

такого, что $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n \geq 0$ и $\|w^l\|^2 = 1$.

Таким образом, в алгебраических терминах решение этой задачи тесно связано с проблемой собственных значений и нахождения ранга корреляционной матрицы; в геометрическом смысле – это задача перехода в пространство более низкой размерности с минимальной потерей информации; в статистическом смысле – это задача нахождения множества ортогональных векторов в пространстве входов, принимающих на себя максимальную вариацию данных, и, наконец, в алгоритмическом смысле – это задача последовательного определения (выделения) множества собственных векторов w^1, w^2, \dots, w^m путем оптимизации каждого из локальных функционалов, образующих глобальный критерий

$$I_w(k) = \frac{1}{k} \sum_{l=1}^m \sum_{u=1}^k (x^{CT}(u) w^l)^2$$

при ограничениях

$$\begin{cases} w^{lT} w^p = 0, \text{ при } l \neq p \\ w^{lT} w^p = 1. \end{cases}$$

Первая главная компонента, несущая максимум информации о контролируемом сигнале, может быть найдена путем максимизации локального критерия

$$I_w^1(k) = \frac{1}{k} \sum_{u=1}^k (x^C(u) w^1)^2$$

с помощью стандартного метода неопределенных множителей Лагранжа.

Далее из каждого вектора $x^C(u)$ вычитается его проекция на первую главную компоненту и вычисляется первая главная компонента остатков, являющаяся второй главной компонентой исходных данных и ортонормальная к первой.

Третья главная компонента вычисляется путем проектирования каждого исходного вектора на первую и вторую главные компоненты, вычитание этой проекции из каждого $x^C(u)$ и нахождения первой главной компоненты полученных остатков, являющейся третьей компонентой исходных данных. Остальные главные компоненты вычисляются рекурсивно согласно описанной стратегии.

К настоящему времени разработано достаточно развитое математическое и программное обеспечение для реализации преобразования Карунена-Лоева, объединение одним общим недостатков: необходимостью априорного задания матрицы X фиксированной размерности. Если же данные поступают последовательно в реальном времени, стандартные процедуры факторного анализа становятся неработоспособными.

В связи с этим целесообразно использование рекуррентных процедур реального времени для нахождения собственных векторов матрицы $R(k)$ путем последовательной обработки наблюдений многомерного временного ряда $x(1), x(2), \dots, x(k), x(k+1) \dots$ без вычисления самой корреляционной матрицы.

В [7] описан искусственный нейрон на основе адаптивного линейного ассоциатора для вычисления первой главной компоненты в реальном времени. На рис. 1 произведена схема этого нейрона, модифицированная для решения задачи обнаружения изменения свойств в многомерном сигнале на основе анализа главной компоненты.

Для предварительно центрированных данных алгоритм обучения имеет вид

$$\begin{cases} w^1(k+1) = w^1(k) + \eta(k+1)(x(k+1) - y(k)w^1(k))y(k+1), \\ y(k+1) = x^T(k+1)w^1(k), w^1(0) \neq 0, y'(1) = x^T(1)w^1(0), \end{cases} \quad (2)$$

где $\eta(k+1)$ – параметр шага настройки, выбираемый достаточно малым для обеспечения устойчивой работы алгоритма. Алгоритм (2) обеспечивает нормирование вектора $w^1(k)$

$$\|w^1(k)\|^2 = 1,$$

сам вектор $w^1(k)$ является собственным вектором матрицы $R(k)$, соответствующим максимальному собственному значению, а выходной сигнал $y(k)$ характеризуется максимально возможной диспер-

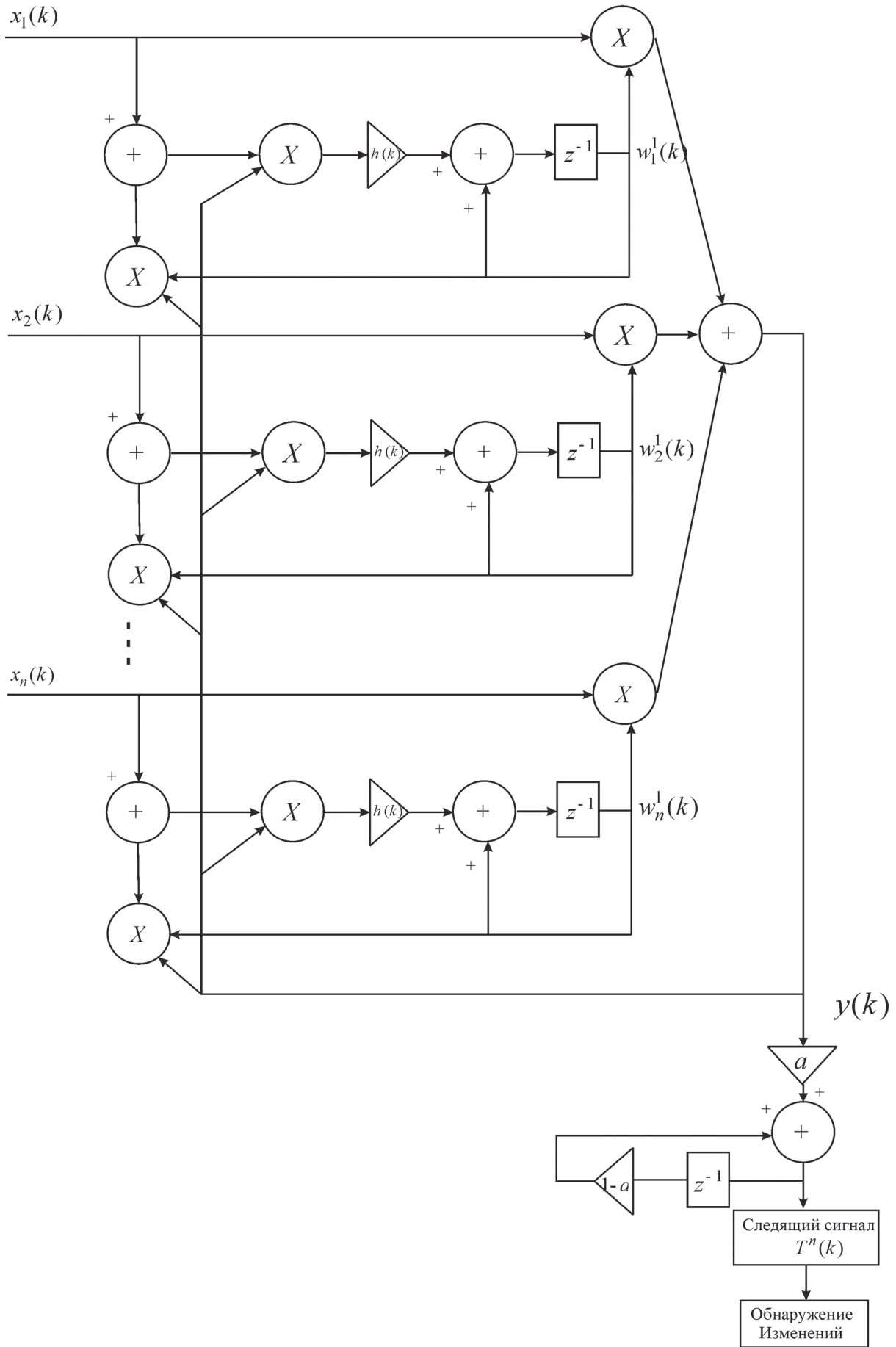


Рис. 1. Модифицированный нейрон для обнаружения изменения свойств главной компоненты многомерного временного ряда

сией, то есть содержит максимум информации о многомерном входном сигнале $x(k)$.

Таким образом, в результате преобразований получаем одномерный временной ряд, проанализировать который уже значительно легче.

2. Результаты эксперимента

В качестве исходных данных был взят небольшой видеофрагмент из 550 кадров. На начальном этапе было необходимо получить результаты сегментации исходных данных. В качестве алгоритма сегментации был выбран JSeg, и проведена сегментация с различными параметрами чувствительности (Scales, Quantresh, Merge), для того чтобы определить насколько влияют эти настраиваемые параметры на общие результаты обнаружения изменения свойств многомерного временного ряда. Пример исходного изображения и результаты его сегментации приведены на рис. 2.



Рис. 2. Пример исходного кадра и результаты его сегментации

Далее для формирования матрицы (1) для каждого сегмента каждого кадра был найден ряд характеристик, таких как площадь сегментов, их периметр, площадь описанной окружности, угол наклона сегмента, характеристик на базе вписанных и описанных эллипсов и так далее [8].

Общее число параметров равно 11. Таким образом, для каждого кадра мы получили матрицу размерности $k \times n$, где k – число сегментов, а n – число находимых характеристик.

После произведенных преобразований были построены графики изменения свойств ряда на различных параметрах сегментации (рис. 3). Scales менялся от 1 до 3, Quantresh в двух случаях равен 200 и в одном 600, а Merge был неизменным и равнялся 0,2. Как видно на представленном графике в районе диапазона кадров 40–60, было явное изменение характеристик. В действительности в это время появился посторонний объект, который, пробыв некоторое время в кадре, исчез. Также следует отметить общую тенденцию всех трендов к уменьшению полученных значений, которое происходило примерно с 250 кадра и достаточно резко прекратилось, что привело к возвращению значений на привычный уровень примерно в районе 310 кадра. В действительности в диапазоне от 25 до 310 в кадре по его краю образовалась новая область, которая постепенно увеличивалась из-за плавного изменения положения камеры. Резкое же возвращение произошло из-за практически мгновенного изменения положения видеокамеры, и, соответственно, резкого пропадания данной области из поля зрения. Таким образом, видно, что за время анализа произошло как минимум два изменения сцены в указанных выше диапазонах, и мы можем указать границы этих диапазонов как ключевые кадры для дальнейшего облечения их поиска и анализа.

Выводы

Таким образом, можно сделать вывод о том, что в результате данной работы промоделирован подход, позволяющий отслеживать изменения свойств многомерных временных рядов на базе метода главных компонент, что позволяет выявлять моменты изменения сцены и отмечать отдельные кадры в качестве ключевых. Следует отметить, что

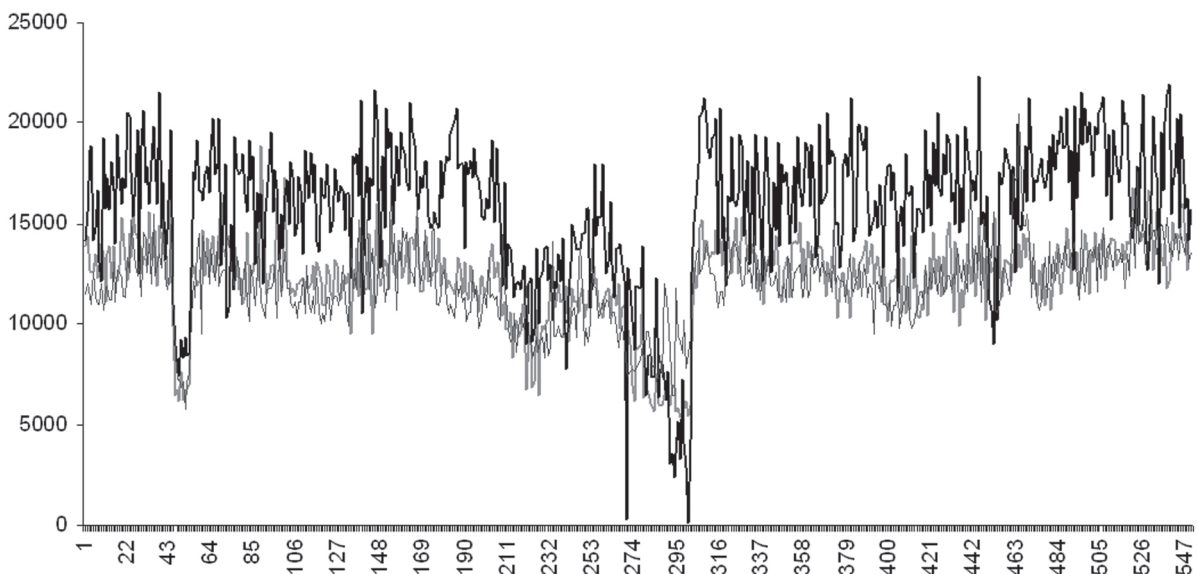


Рис. 3. Результаты эксперимента с различными параметрами сегментации

параметры сегментации, используемые в JSeg, оказывают влияние на разброс значений, но видно, что общий тренд соблюдается для всех параметров. При этом следует отметить, что небольшие изменения, которые происходили практически на протяжении всего видеофрагмента, не захватывались, то есть данный подход применим только в случае явных изменений сцены, в остальных ситуациях, то есть когда нужно отследить именно мелкие изменения, данный подход может дать весьма неоднозначные результаты. В дальнейшем планируются исследования, касающиеся повышения быстродействия и устойчивости к помехам для данного подхода.

Список литературы: 1. *Ширлев, А.Н.* Статистический последовательный анализ [Текст] / А.Н. Ширлев. – Москва: Наука, 1976. – 272 с. 2. *Никифоров, И.В.* Последовательное обнаружение изменения свойств временных рядов [Текст] / И.В. Никифоров. – Москва: Наука, 1983. – 198 с. 3. *Бассвиль, М.* Обнаружение изменений свойств сигналов и динамических систем [Текст] / М. Бассвиль, А. Вилски, А. Банвенист и др. – М: Мир, 1989. – 278 с. 4. *Гребенюк, Е.А.* Методы анализа нестационарных временных рядов с неявными изменениями свойств [Текст] / Е.А. Гребенюк // Автоматика и телемеханика. – 2005. – №12. – С. 3-30. 5. *Маликаускас, В.* Оценка моментов изменения свойств многомерных случайных последовательностей [Текст] / В. Маликаускас // Статистические проблемы управления. – 1988. – Вып.83. – С. 199-204. 6. *Иберла К.* Факторный анализ / К. Иберла. – Москва: Статистика, 1980. – 398 с.

7. *Cichocki A., Unberhauen R.* Neural Networks for Optimization and Signal Processing. – Stuttgart: Teubner, 1993. – 526 p. 8. *V. Kindratenko,* “Development and Application of Image Analysis Techniques for Identification and Classification of Microscopic Particles”, Ph.D. Thesis, University of Antwerp, Belgium, 1997.

Поступила в редколлегию 19.03.2010 г.

УДК 519.673

Пошук змін в відеоданих за допомогою аналізу результатів сегментації / С.В. Машталір, С.В. Постульга, К.С. Щербінін // Біоніка інтелекту: наук.-техн. журнал. – 2010. – № 1 (72). – С. 65–69.

В роботі запропоновано та проаналізовано підхід до виявлення змін багатовимірних часових рядів на прикладі обробки результатів сегментації відео даних. Для виявлення змін запропоновано переходити до одновимірних рядів за допомогою перетворення Карунена-Лоева.

Бібліогр.: 8 найм.

UDK 519.673

Changes detection in videodata by result of segmentation analysis / S.V. Mashtalir, S.V. Postulga, K.S. Scherbinin // Bionics of Intelligence: Sci. Mag. – 2010. – № 1 (72). – P. 65–69.

Approach to changes detection in multivariate time-series for example, the processing of video data segmentation, are proposed and analyzed. Conversions to a one-dimensional series by Karhunen-Loev transformation to changes detection are proposed.

Ref.: 8 items.

УДК 004.89



ИССЛЕДОВАНИЕ ИММУННЫХ ОПЕРАТОРОВ В ЗАДАЧЕ КЛАСТЕРИЗАЦИИ ОБЪЕКТОВ

Н.М. Кораблев¹, А.А. Фомичев²¹ ХНУРЭ, г. Харьков, Украина, korablev@kture.kharkov.ua² ХНУРЭ, г. Харьков, Украина, alexandros_1985@mail.ru

В данной работе исследуются иммунные операторы, которые используются для решения задачи кластеризации объектов. На этапе обучения применяется приоритетное последовательное клонирование с конкурентно-целевым отбором клонов, и предложен метод разброса объектов в определяемых ограниченных областях. Спецификой работы иммунных операторов является использование единого критерия отбора объектов, применяемого как на этапе обучения, так и на этапе формирования кластеров.

ИСКУССТВЕННЫЕ ИММУННЫЕ СИСТЕМЫ, ПРИОРИТЕТНОЕ ПОСЛЕДОВАТЕЛЬНОЕ КЛОНИРОВАНИЕ, КОНКУРЕНТНО-ЦЕЛЕВОЙ ОТБОР КЛОНОВ, ОГРАНИЧЕННЫЕ ОБЛАСТИ РАЗБРОСА

Введение

В настоящее время в информационных технологиях используются подходы, позволяющие проводить интеллектуальную обработку и систематизацию данных. Одним из методов систематизации является кластеризация объектов, основной задачей которой является разбиение некоторого множества объектов по ряду признаков на кластеры [1-3]. Среди биологических принципов кластеризации можно выделить искусственные иммунные системы [4-7], рабочими элементами которых являются популяции антигенов и антител.

В настоящее время разработаны иммунные алгоритмы кластеризации, которые на некоторых этапах работы используют одинаковые подходы для решения задач клонирования, отбора и дополнительного разброса [6, 7]. В этих алгоритмах на этапах отбора антител и клонов производится большое количество избыточных вычислений. Это связано с тем, что клонируемые антитела конкурируют между собой за количество клонов. Такой способ организации клонирования не обеспечивает максимального покрытия антигенов, что увеличивает количество шагов в цикле их восстановления.

В предлагаемом алгоритме реализуются новые подходы, используемые в работе иммунных операторов клонирования и отбора клонов. Для отбора объектов используется единый критерий, основанный на плотности распределения исходной популяции антигенов, предложен новый способ организации клонирования, при котором каждое антитело создает максимально возможное количество клонов, в результате чего восстанавливается большее количество антигенов. Оператор отбора клонов реализуется на основе использования конкурентно-целевого отбора. Оператор дополнительного разброса реализуется путем разброса антител в определяемых ограниченных областях. Предложен также подход к определению центров

кластеров и их границ. Использование этих методов сокращает избыточные вычисления и способствует ускорению работы иммунного алгоритма.

1. Постановка задачи

Дано множество объектов $M\{m_1; \dots; m_n\}$, $n = \overline{1, N}$, каждый из которых описывается набором признаков $S\{s_1; \dots; s_k\}$, $k = \overline{1, K}$ (в простейшем случае это координаты объекта). Эти объекты представляются в виде популяции антигенов $AG(ag_1; \dots; ag_n)$. В качестве меры близости между объектами m_i и m_j используется критерий аффинности Af_{ij} [4-7]:

$$Af_{ij} = (1 + d_{ij})^{-1}, \quad (1)$$

$$d_{ij} = \sqrt{\sum_{m=1}^k (s_{im} - s_{jm})^2}, \quad (2)$$

где d_{ij} – евклидово расстояние между объектами; s_{im} – m -й признак у i -го объекта.

Необходимо разработать ИИС-алгоритм кластеризации исходного множества объектов $M\{m_1; \dots; m_n\}$, использующий в качестве основной меры близости объектов критерий (1), позволяющий производить их кластеризацию.

2. Алгоритм кластеризации

Работу иммунного алгоритма кластеризации можно разделить на два основных этапа: 1) обучение (восстановление популяции антигенов); 2) формирование кластеров.

В соответствии с этим процесс кластеризации может быть описан следующим образом:

$$\begin{aligned} fclust(ag_1; \dots; ag_n) &= result(reconstr(ab_1; \dots; ab_n), \\ &clust(ab'_1; \dots; ab'_n)), \\ (ab'_1; \dots; ab'_n) &= reconstr(ab_1; \dots; ab_n), \end{aligned} \quad (3)$$

где $reconstr(ab_1; \dots; ab_n)$ – восстановление антигенов (обучение) путем применения операторов отбора, клонирования, мутации и старения к популяции

антител $AB(ab_1; \dots; ab_n)$, а $clust(ab_1'; \dots; ab_n')$ – формирование кластеров. Тогда процесс восстановления антигенов может быть представлен следующим образом:

$$\begin{aligned} reconstr(ag_1; \dots; ag_n) &= reconstr(sel(ab_1; \dots; ab_n), \\ &clon(ab_1'; \dots; ab_n'), ageing(ab_1''; \dots; ab_n''), \\ &dispers(ab_1; \dots; ab_n)), \\ (ab_1'; \dots; ab_n') &= sel(ab_1; \dots; ab_n), \\ (ab_1''; \dots; ab_n'') &= clon(ab_1'; \dots; ab_n'), \end{aligned} \quad (4)$$

где $sel(ab_1; \dots; ab_n)$ – оператор отбора антител; $clon(ab_1'; \dots; ab_n')$ – оператор клонирования, мутации и отбора клонов; $ageing(ab_1''; \dots; ab_n'')$ – оператор старения антител; $dispers(ab_1; \dots; ab_n)$ – оператор дополнительного разброса объектов.

В работе оператора отбора антител $sel(ab_1; \dots; ab_n)$ (первичного и после дополнительного разброса) используется критерий средней аффинности антигенов $CSel$ [9]:

$$CSel = \frac{\sum_{i=1}^n AF_{iAG}}{n}, \quad (5)$$

где AF_{iAG} – средняя аффинность антигена ag_i со всеми антигенами выборки $AG(ag_1; \dots; ag_n)$, которая определяется следующим образом:

$$AF_{iAG} = \frac{\sum_{j=1}^n af_{ij}}{n}, \quad (6)$$

где af_{ij} – аффинность (1) антигена ag_i и антигена ag_j . Это упрощает процедуру отбора и повышает вероятность восстановления антигенов на последующих этапах работы иммунного алгоритма.

Антитело проходит отбор в том случае, если его средняя аффинность с антигенами выборки, определяемая по (6), удовлетворяет условию:

$$Af_{iAG} \geq 98\% \times CSel. \quad (7)$$

Оператор клонирования $clon(ab_1'; \dots; ab_n')$ в рассматриваемом алгоритме использует приоритетное последовательное клонирование с конкурентно-целевым отбором клонов. Непосредственно процедура клонирования и отбора клонов может быть представлена в виде следующей последовательности операторов:

$$\begin{aligned} clon(ab_1; \dots; ab_n) &= clon(ccreate(ab_1; \dots; ab_n), \\ &cmut(ab_1'; \dots; ab_n'), csel(ab_1''; \dots; ab_n'')), \\ (ab_1'; \dots; ab_n') &= create(ab_1; \dots; ab_n), \\ (ab_1''; \dots; ab_n'') &= mut(ab_1'; \dots; ab_n'), \end{aligned} \quad (8)$$

где $ccreate(ab_1; \dots; ab_n)$ – оператор создания клонов; $cmut(ab_1'; \dots; ab_n')$ – оператор мутации клеток, а $csel(ab_1''; \dots; ab_n'')$ – оператор отбора клонов.

В [6, 7] клонирование выполняется одинаковым образом и заключается в том, что для каждого клонируемого антитела отводится некоторое количество клонов (от их общего числа), определяемое в зависимости от его параметров (средней аффинности с антигенами). Особенностью такой организации работы оператора клонирования является создание клонов одновременно для всей популяции антител. При таком распределении клонов каждый объект получает небольшое количество клонов, что уменьшает вероятность восстановления исходных антигенов. Это приводит к тому, что на восстановление антигенов требуется большое количество поколений антител и, следовательно, большие затраты времени. В предлагаемом алгоритме реализован метод приоритетного последовательного клонирования, суть которого заключается в том, что для каждого клонируемого объекта независимо от его параметров выделяется максимально возможное количество клонов. Создание, мутация и отбор клонов происходит последовательно для каждого клонируемого антитела, а не для всей популяции одновременно, как в [6, 7]. Благодаря этому растет площадь разброса клеток и увеличивается вероятность восстановления антигенов, однако при этом большое значение приобретает процедура отбора клонов. Суть предлагаемого метода конкурентно-целевого отбора заключается в том, что для каждого клонируемого объекта (антитела) определяется расстояние поиска целевых антигенов r :

$$r = k \times \sqrt{\frac{width \times height}{n}}, \quad (9)$$

где $width$ – ширина; $height$ – высота области кластеризации; n – количество антигенов; k – коэффициент увеличения области поиска. Для всех клонов объекта после их создания и мутации определяются аффинности с целевыми антигенами, находящимися в области поиска r , после чего для каждого клона по максимальному значению аффинности определяется целевой антиген. Отбор клонов осуществляется за счет конкуренции клеток по целевому признаку, то есть из всего множества клонов, имеющих общий целевой антиген, отбирается тот, аффинность которого к данному антигену больше. При такой организации клонирования может возникнуть ситуация, при которой объектам для клонирования может быть недостаточно клонов. Для решения данной проблемы используются приоритеты. Это выражается в том, что объекты, которым не хватило клонов, на следующем шаге

будут обладать повышенным приоритетом и будут клонироваться в первую очередь.

После выполнения операторов отбора, клонирования и старения популяции антител остается некоторое количество неиспользованных антител. Для соблюдения равенства популяций антигенов и антител необходимо производить дополнительный разброс. В работе оператора дополнительного разброса предлагается формирование ограниченных областей разброса. В иммунных алгоритмах [6, 7] дополнительный разброс производится случайным образом. Вероятность восстановления антигенов при такой организации разброса невелика. В предлагаемом алгоритме кластеризации при организации дополнительного разброса использовался подход, ориентированный на формирование ограниченных областей с использованием контрольных точек.

Перед началом работы оператора восстановления выборки все пространство обработки данных разбивается на области дополнительного разброса со стороной r (9). После этого в каждой полученной области дополнительного разброса определяются контрольные точки, значения аффинностей в которых будут характеризовать область. Таким образом, на каждую область отводится всего пять точек, в которых при дополнительном разбросе устанавливаются антитела. Затем по аффинностям контрольных точек с восстанавливаемой популяцией антигенов вычисляется средняя аффинность всей области дополнительного разброса. В том случае, если аффинность области дополнительного разброса не удовлетворяет условию (7), область исключается из списка областей дополнительного разброса и не учитывается при разбросе антител. Таким образом, устанавливаются контуры больших областей дополнительного разброса, располагающихся на скоплении антигенов, а удаленные от таких скоплений области не будут учитываться.

На рис. 1 приведен пример формирования областей дополнительного разброса в местах скопления антигенов предлагаемым методом.

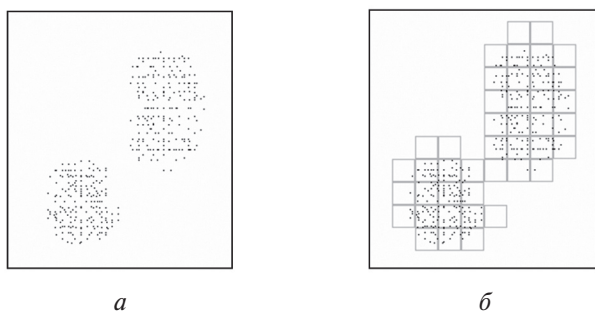


Рис. 1. Организация дополнительного разброса:
 а – исходное расположение антигенов;
 б – области дополнительного разброса

Определение исходных кластеров – важнейшая задача, решаемая при кластеризации данных. В рас-

сматриваемом алгоритме предлагается новый подход к решению данной задачи. Процесс формирования кластеров можно представить в виде следующей последовательности шагов:

Шаг 1. Из популяции $AB(ab_1; \dots; ab_n)$ с неопределенными кластерами, определяется антитело с максимальной аффинностью Af_{iAG} , которое будет являться условным центром нового кластера.

Шаг 2. Из популяции $AB(ab_1; \dots; ab_n)$ с неопределенными кластерами, отбираются объекты, расположенные на расстоянии d (10) от центра кластера:

$$d = Af_{iAG} \times \sqrt{\frac{width \times height}{n}}, \quad (10)$$

если таковых нет – кластер удаляется.

Шаг 3. Для каждого антитела, входящего в кластер, определяется расстояние d (10) и осуществляется процесс поиска соседних антител (происходит расширение кластера).

Шаг 4. Возвращение к Шагу 1 до тех пор, пока не останется некластеризованных антител, либо дальнейшее формирование новых кластеров будет невозможно.

Кластеризация антител происходит после восстановления большинства антигенов (то есть достижения некоторого заданного порога восстановления клеток).

Кластеризация популяции исходных антигенов производится по результатам кластеризации последней популяции антител. Для каждого невосстановленного антигена вычисляются аффинности со всеми существующими кластерами, после чего оно относится к ближайшему кластеру, то есть к кластеру, аффинность по отношению к центру которого выше.

Алгоритм кластеризации объектов можно представить в виде следующей последовательности:

1. Формирование начальной популяции антител.
2. Определение критерия отбора антител.
3. Определение областей дополнительного разброса.
4. Отбор антител по условию (7).
5. Цикл восстановления и кластеризации антител:
 - 5.1. Отбор антител, полученных в результате дополнительного разброса.
 - 5.2. Последовательное клонирование антител по приоритету.
 - 5.3. Удаление клонированных антител, которые не восстановили антигены.
 - 5.6. Дополнительный разброс антител.
 - 5.7. Проверка критерия восстановления антител. В случае его достижения переход к п.5.8., иначе возвращение к п. 5.1.
 - 5.8. Определение параметров кластеров антител.

5.8.1. Определение центра кластера.

5.8.2. Определение исходных границ кластера.

5.8.3. Расширение кластера.

5.8.4. Возврат к п. 5.8.1. до тех пор, пока возможно формирование новых кластеров.

6. Формирование кластеров антигенов по результатам кластеризации последней популяции антител.

В результате работы будет получено множество кластеров $CL(c_1; \dots; c_m)$, сформированных из популяции исходных антигенов $AG(ag_1; \dots; ag_n)$.

3. Результаты экспериментальных исследований

Тестирование иммунного алгоритма кластеризации производилось на изображении размером 270×240 точек. Выборка состояла из 1000 антигенов, сформированных случайным образом в четырех областях разброса.

За счет последовательной организации клонирования антитела уже на первой итерации восстановили более 25% антигенов. В данном примере порог восстановления был принят равным 80%. Так, на 5-й итерации алгоритма начался процесс кластеризации антител. Для восстановления 99,6% антигенов потребовалось 7 итераций. В результате кластеризации последней популяции антител и определения кластеров антигенов были сформированы 4 кластера (рис. 2).

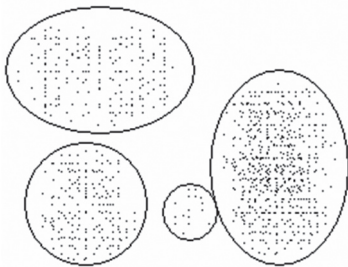


Рис. 2. Результаты кластеризации

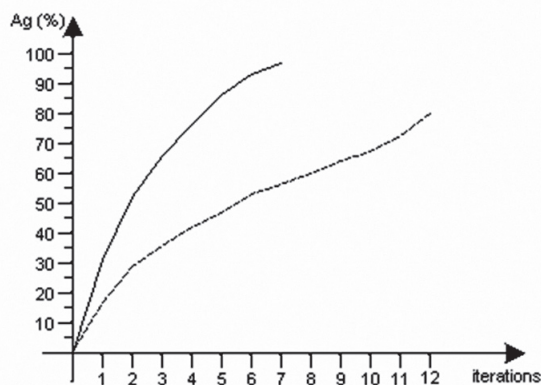


Рис. 3. Графики восстановления антигенов с использованием различных алгоритмов

На рис. 3 приведены графики восстановления антигенов с помощью предлагаемого алгоритма (сплошная линия) и алгоритма, приведенного в [6] (пунктирная линия). Следует отметить, что на-

ибольшее количество случаев восстановления антигенов выборки приходится на первую итерацию. Затем эффективность восстановления клеток снижается от итерации к итерации с 20% на второй – до 8,5% на седьмой. Это обусловливается уменьшением количества возможных клонов на каждой последующей итерации.

Как видно по результатам тестирования, предлагаемый алгоритм практически полностью восстанавливает обучающую выборку за небольшое количество итераций, в то время как в соответствии с алгоритмом [6] для восстановления исходной популяции антигенов потребовалось в два раза больше поколений.

Выводы

В работе исследовались иммунные операторы, используемые для кластеризации объектов. В работе оператора клонирования было предложено использование приоритетного последовательного клонирования с конкурентно-целевым отбором клонов, что значительно ускорило процесс восстановления исходных антигенов. Для обеспечения работы оператора отбора антител было предложено использование критерия средней аффинности, что упростило процедуру отбора и повысило ее точность. В операторе дополнительного разброса антител был предложен метод разброса в определяемых ограниченных областях, что увеличило точность дополнительного разброса и ускорило процесс восстановления антигенов.

При формировании кластеров объектов был предложен новый подход к определению центров кластеров и их границ, позволяющий ускорить кластеризацию объектов.

Предложенный алгоритм кластеризации, несмотря на некоторую сложность в реализации, обеспечивает быстрое восстановление антигенов и распределение их по кластерам.

Список литературы: 1. A.K. Jain, M.N. Murty, P.J. Flynn – «Data Clustering: A Review», ACM Computing Surveys, 1999, P. 265-323. 2. G. Gan, C. Ma, J. Wu – «Data Clustering: Theory, Algorithms, and Applications», ASA-SIAM, Alexandria, 2007, 488 p. 3. W. Pedrycz, «Knowledge-Based Clustering», Wiley&Sons, 2005, 377 p. 4. Искусственные иммунные системы и их применение [Текст] / под ред. Д. Дасгупты: пер. с англ. А.А. Романюхи. – М.: ФИЗМАТЛИТ, 2006. – 344 с. 5. J. Timmis, T. Knight, L.N. de Castro, E. Hart. «An Overview of Artificial Immune Systems», Natural Computation, Springer, 2004, P. 55-86. 6. Lei Jia, Licai Yang, Qingjie Kong, Shu Lin. «Study of Artificial Immune Clustering Algorithm and Its Applications to Urban Traffic Control», International Journal of Information Technology, Vol.12, No.3, 2006, 9 p. 7. Литвиненко, В.И. Гибридная иммунная сеть для решения задач структурной идентификации [Текст]/ В.И. Литвиненко, П.И. Бидюк, А.А. Фефелов, И.В. Баклан // Нейронные сети. – № 1 – 2005. – С. 143-155.

Поступила в редколлегию 22.03.2010 г.

УДК 004.89

Дослідження імунних операторів в задачі кластеризації об'єктів / М.М. Кораблев, О.О. Фомічов // Біоніка інтелекту: наук.-техн. журнал. – 2010. – № 1 (72). – С. 70–74.

У статті досліджується робота імунних операторів, що використовуються при кластеризації об'єктів. При відновленні популяції антигенів використовується пріоритетне послідовне клонування з конкурентно-цільовим відбором клонів. В роботі оператора додаткового розкиду запропоновано метод розкиду об'єктів у визначених обмежених областях. Для організації відбору антитіл та клонів використовується загальний критерій, що використовується також і при формуванні кластерів.

Л. 3. Бібліогр.: 7 найм.

UDC 004.89

Study of immune operators in problems of clustering objects / N.M. Korabljev, A.A. Fomichev // Bionics of Intelligence: Sci. Mag. – 2010. – № 1 (72). – P. 70–74.

The paper investigates the work of the immune operators used in the clustering of objects. When restoring a population of antigen used priority consistent with the cloning competitive target selection of clones. In this paper, the operator further spread of a method for scatter objects in defined limited areas. To organize the selection of antibodies and clones used the general criterion, also used in the formation of clusters.

Fig. 3. Ref.: 7 items.

УДК 658.012.011.56



ТЕКСТ НА ЕСТЕСТВЕННОМ ЯЗЫКЕ КАК ОБЪЕКТ СТАТИСТИЧЕСКОГО АНАЛИЗА

М.Ю. Крыгин

Украинский языково-информационный фонд НАН Украины, Киев, Украина, maxus@zeos.net

Статья посвящена применению статистических методов к задачам автоматической обработки текста естественного языка. Определен круг задач по исследованиям текста на естественном языке, решаемых статистическими методами. Сформулированы теоретические основы применения статистических методов к обработке текстов, описаны модели, приведены примеры решения некоторых задач.

СТАТИСТИЧЕСКИЙ ПОРТРЕТ ТЕКСТА, ЦЕПЬ МАРКОВА, ВЕРОЯТНОСТЬ ПЕРЕХОДА, ГЕНЕРАТОР ТЕКСТОВ, ЭЛЕМЕНТ ТЕКСТА

Введение

Статистические методы в науке используются там, где есть возможность доступа к большим объемам данных, а также, если требуется исследовать некоторое достаточно распространенное явление на относительно небольшом массиве данных.

Данная работа посвящена разработке методов, позволяющих получить статистические характеристики языковых явлений для того, чтобы, например, уметь идентифицировать их или дать оценку о наличии и функционировании определенного явления в определенном тексте или ином фрагменте языкового материала. В данной работе естественноречевой текст рассматривается нами как конечная последовательность символов из некоторого (конечного) алфавита, определенные фрагменты которой получают в ходе анализа лингвистическую интерпретацию.

Целью статистической обработки естественного языка является математическое описание языковых фактов и явлений, связей между ними, получение набора моделей языка для решения определенных лингвистических задач. Здесь ставится задача определить те языковые явления и факты, которые заслуживают внимания – эта задача, возможно, является одной из самых сложных, поскольку от правильного выбора объектов исследования зависит дальнейший результат работы. В наших исследованиях мы выделяем несколько уровней языка: знаковый, фонетический, лексический, грамматический и так далее.

1. Объекты исследования

Исследование языка на уровне знаковой системы преследует цели вычисления распределений символов и их комбинаций в тексте, выявления «запрещенных» сочетаний, нахождение которых в тексте означает ошибку, моделирования языка на символическом уровне.

Исследование фонетической системы дает, в частности, ответ на вопрос, насколько фонетическая система конкретного языка отвечает, например, его орфографической системе.

Достаточно сложными и разнообразными являются статистические исследования лексической и грамматической систем языка. Эти исследования дают возможность не только сгруппировать слова по частотам и получить частотный словарь слов языка, но и исследовать сочетаемость лексических единиц и грамматических форм, приблизить автоматическое выделение устойчивых словосочетаний (например фразеологизмов и терминов) из текста, реализовать (с определенной точностью) автоматический синтаксический анализ и автоматическое снятие омонимии.

Основным массивом данных, на котором мы проводим исследования, является корпус текстов [4]. На массивах корпуса можно исследовать такие явления как грамматическая и лексическая омонимия, установить свойства текста, которые являются характерными, например для конкретного автора. Приведение текста к форме фонетической транскрипции дает возможность провести достаточно полное статистическое исследование фонетической системы языка. В нашей работе мы предполагаем, что тексты корпуса не содержат ошибок, пропусков и полностью совпадают с оригиналами (то есть мы работаем с выправленными текстами). Тексты корпуса подлежат специальной предварительной разметке согласно целям исследования, которому они подвергаются.

Кроме текстов корпуса, в отдельных случаях исследования проводились на Грамматическом словаре украинского языка [1], в котором представлены все словоформы, имеющие место в языке.

2. Статистический портрет текста

В наших исследованиях используется понятие статистического портрета текста, под которым мы понимаем определенным образом сформированную совокупность статистических признаков текста, полученных путем статистической его обработки по неким заранее установленным принципам.

Множество таких признаков состоит из подмножеств, которые относятся к различным уровням (аспектам) языковой системы и предоставляют данные о статистике:

- знаковой системы, в том числе для фонетического (транскрибированного) варианта текста;
- лексической и грамматической систем;
- семантической системы;
- синтаксической системы.

И так далее.

Определим формально понятие статистического портрета текста.

Пусть у нас имеется некоторый текст T , для которого определена некая система отображений:

$$\sigma_{\alpha} : T \rightarrow T_{\alpha}, \text{ где } T_{\alpha} = \bigcup_i T_{\alpha}^i,$$

где T_{α}^i – набор параметров, отображающих определенное статистическое свойство текста, например распределение символов текста, вероятности перехода или распределение словоформ; T_{α} – статистический портрет текста T . Он представляет собой объединение наборов статистических параметров для всех свойств текста, которые исследуются.

Нижайший уровень исследования текста производится на его знаковой системе. Исследования статистических свойств языка на уровне знаковой системы явились одними из первых в статистике языка.

Обозначим через U_M обобщенный алфавит языка M , то есть конечное множество всех символов, которые встречаются при графическом (письменном) отображении текстов данного языка. Это множество является более широким, чем алфавит языка за счет того, что в него входят, кроме букв алфавита, еще и пробел, знаки пунктуации, дефис, специальные символы (включая символы чисел). Таким образом, U_M имеет такую структуру:

$$U_M = B_M \cup P_M \cup S_M \cup \pi,$$

где B_M – обычный алфавит языка M ; P_M – множество знаков пунктуации:

$P_M = \{p_1 = '.', p_2 = ',', p_3 = ';', p_4 = ':', p_5 = '...', p_6 = '!', p_7 = '?', p_8 = '- ', p_9 = '"', p_{10} = ')', p_{11} = '('\}$; S_M – множество специальных знаков; через π обозначен пробел.

Для украинского языка B_M имеет вид:

$$B_{УКР} = \{b_1 = 'А', b_2 = 'Б', b_3 = 'В', b_4 = 'Г', b_5 = 'Г', b_6 = 'Д', \dots, b_{30} = 'Щ', b_{31} = 'Ь', b_{32} = 'Ю', b_{33} = 'Я', b_{34} = 'апостроф'\}.$$

Объединение B_M , дефиса и пробела обозначим через A_M :

$$A_M = B_M \cup \alpha \cup \{-\}.$$

Будем рассматривать текст как линейную последовательность элементов: $T = x_1 x_2 \dots x_N$, N – длина текста в определенных элементах. Элементом текста T в зависимости от уровня его рассмотрения может быть символ алфавита языка, лексема, граммема, семантическое состояние, фразеоло-

гическая или синтаксическая конструкция и тому подобное. На тексте T определим распределение его элементов как вероятность встретить определенный элемент в тексте T , то есть отношение количества вхождений данного элемента в текст к числу всех элементов данного типа в тексте; аналогично зададим распределение сочетаний по j элементов текста как количество вхождений данного сочетания в текст к $N - j$. Определим вероятность перехода из элемента x в элемент y как отношение количества сочетаний этих элементов в тексте к количеству всех сочетаний элементов в тексте по два, первый из которых является элементом x . Аналогично определим вероятность перехода из последовательности элементов $\{x_1 x_2 \dots x_n\}$ в элемент y как отношение количества вхождений в текст сочетаний элементов $\{x_1 x_2 \dots x_n y$ (элемент y – фиксирован)} к количеству вхождений сочетаний элементов $\{x_1 x_2 \dots x_n x$ (x пробегает все множество возможных элементов текста)}. На множестве текстов языка M зададим семейство отображений $\sigma_{el}^{\lambda} : T \rightarrow T_{el}, \lambda = 1, 2, \dots, 10$ следующим образом:

$$\sigma_{el}^1 : T \rightarrow T_{el}^1 = \{(a); P(a)\}$$

$$\sigma_{el}^2 : T \rightarrow T_{el}^2 = \{(a_1 a_2); P(a_1 a_2)\}$$

...

$$\sigma_{el}^{10} : T \rightarrow T_{el}^{10} = \{(a_1 a_2 \dots a_{10}); P(a_1 a_2 \dots a_{10})\}$$

и семейство отображений $\sigma_{mark}^{\lambda} : T \rightarrow T_{mark}, \lambda = 1, 2, \dots, 10$ – следующим образом:

$$\sigma_{mark}^1 : T \rightarrow T_{mark}^1 = \{P(x_i = a_k / x_{i-1} = a_m)\},$$

i – порядковый номер элемента x в тексте;

$$\sigma_{mark}^2 : T \rightarrow T_{mark}^2 = \{P(x_i = a_k / x_{i-1} = a_m, x_{i-2} = a_n)\};$$

i – порядковый номер элемента x в тексте;

...

$$\sigma_{mark}^{10} : T \rightarrow T_{mark}^{10} = \{P(x_i = a_k / x_{i-1} = a_m, x_{i-2} = a_n, \dots,$$

$x_{i-10} = a_l)\}, i$ – порядковый номер элемента x в тексте}.

Число 10 было выбрано как максимальная длина исследуемых последовательностей элементов текста с тем, чтобы покрыть среднюю длину слова (при рассмотрении знаковой системы текста), а также длину словосочетаний, синтаксических и грамматических конструкций.

Статистический портрет текста представляется в виде объединения:

$$T_{\alpha} = \bigcup_i (T_{el}^i \cup T_{mark}^i),$$

где T_{el}^i представляют собой распределения сочетаний элементов текста по i ; T_{mark}^i представляют собой вероятности перехода, описанные выше.

При рассмотрении знаковой системы текста элементами x являются символы алфавита языка A_M . Для лексического же уровня элементами текста являются уже определенные словоформы, а множеством значений элементов текста — словарь словоформ D_M языка M . $T = w_1 w_2 \dots w_{i-1} w_i w_{i+1} \dots w_N$, $w_i \in D_M$, N — длина текста в словах, D_M — расширенный словарь языка M , который содержит все возможные в языке словоформы. Для уровня грамматической системы элементами текста являются так называемые грамматические состояния, формальными репрезентантами которых служат грамматические значения, маркированные соответствующими грамматическими категориями. При этом текст представляется в виде:

$$T = g_1 g_2, \dots, g_{i-1}, g_i, g_{i+1}, \dots, g_N,$$

$g_i \in GD_M$, N — длина текста в словах, GD_M — грамматический словарь языка M , который для каждой словоформы задает ее грамматическое значение $g = (p, f)$, где p — часть речи, f — грамматическое значение слова.

Для получения грамматических значений слов применяются специальные функции [5], разработанные в Украинском языково-информационном фонде НАН Украины. Основную проблему в данном исследовании представляет явление грамматической омонимии и, как следствие, невозможность автоматически однозначно присвоить словоформе ее грамматическое значение. Неомонимичные же словоформы получают грамматическое значение автоматически. Омонимичным словоформам присваивают грамматические значения в полуавтоматическом режиме: для каждой такой словоформы предлагается выбрать из списка возможных вариантов часть речи, которой они принадлежат, а затем грамматическое значение внутри части речи.

Использование статистических методов позволяет также разработать модель семантики и предложить методику построения семантических структур. Эталоном, который используется для построения семантических структур, в данном исследовании является словарь украинского языка (СУЯ) [2]. Для построения семантической модели текста каждому слову приписывается признак $s = (w_0, w_{num})$, w_0 — начальная форма слова, w_{num} — номер семантического значения в СУЯ. Таким образом, текст T можно представить в виде последовательности признаков: $T^{sem} = s_1 s_2 \dots s_N$.

Статистические исследования проводились на материале корпуса текстов [4] и грамматического словаря украинского языка [1] объемом 3,2 млн словоформ. Ввиду того, что полученные данные довольно громоздки и получение их не являлось целью исследования, приводить их здесь не будем. Приведем лишь несколько заключений, которые были сделаны по результатам статисти-

ческого анализа. Грамматический словарь для статистических исследований был представлен в виде последовательности всех зафиксированных в нем словоформ, разделенных пробелами, например: «... абазин абазина абазинів абазину абазина абазином ...». В приведенных ниже результатах учтены лишь те сочетания символов, которые не содержат пробела нигде, кроме начальной и конечной позиции. Исключение последовательностей символов с пробелом посередине связано с тем, что они не отображают свойство языка в целом, так как являются частями соседних словоформ грамматического словаря и не встречаются в естественных текстах в таком виде.

Таблица 1

Статистика буквосочетаний в Грамматическом словаре украинского языка

Длина сочетания (в символах)	Количество уникальных сочетаний, найденных в словаре	Комбинаторно возможное количество уникальных сочетаний	% реализации от теоретически возможного
1	35	35	100
2	1037	1225	84,65
3	13781	42875	32,14
4	93774	1500625	6,25
5	354544	52521875	0,68
6	850049	1838265625	0,05
7	1524535	64339296875	0,0024
8	2222461	2251875390625	0,000099
9	2740430	78815638671875	0,0000035
10	2921764	2758547353515625	0,0000001

Из приведенной таблицы видно, что количество уникальных буквосочетаний, найденных в грамматическом словаре, существенно зависит от ранга (длины) буквосочетания: при увеличении ранга буквосочетаний растет количество уникальных буквосочетаний, причем наиболее существенный рост наблюдается при переходах от ранга 1 до 7, то есть до достижения средней длины слова украинского языка, далее рост незначителен. Количество уникальных буквосочетаний совпадает с комбинаторно возможным только для ранга 1, также оно остается достаточно высоким для ранга 2, для ранга 3 — существенно ниже, а далее наблюдается резкий спад.

В грамматическом словаре почти полностью представлена вся возможная лексика украинского языка и все возможные комбинации символов в пределах слов, однако он не в полной мере отражает статистические свойства речи вследствие того, что там представлены абсолютно все допустимые грамматической системой словоформы, в том числе и редкоупотребляемые. То есть на основе грамматического словаря можно получить все возможные в естественном языке комбинации символов,

но нельзя достоверно оценить их частотность. Для исследования символьной системы на частотность использовался массив текстов из корпуса [4].

Таблица 2

Зависимость между длиной сочетаний и их количеством с абсолютной частотой > 1 (для Грамматического словаря)

Длина сочетания									
1	2	3	4	5	6	7	8	9	10
100	99,8	99,4	98,9	96,6	92,6	86,4	78	66,8	53,7
Среди них с абсолютной частотой больше 1 (в %)									

Таблица 3

Зависимость между длиной сочетаний и их количеством с абсолютной частотой > 1 (для текста)

Длина сочетания									
1	2	3	4	5	6	7	8	9	10
100	98,72	95,96	86,64	60,49	38,80	21,28	10,04	3,8	0,59
Среди них с абсолютной частотой больше 1 (в %)									

Из приведенных таблиц видно, что в грамматическом словаре повторяемость комбинаций символов выше. Это объясняется тем, что в словаре представлены полные парадигмы для всех слов языка, обеспечивающие наличие последовательностей символов, отвечающих корню слова, от 7 до 20 и даже более раз; кроме того, разные слова могут иметь одинаковые словоизменительные морфемы.

На базе полученных T_{zn} был построен генератор текстов, работающий по законам распределения символов в украинском языке:

$$G : T_{zn} \rightarrow T^{gen}, G \in \{G_0, G_1, G_2, G_N\}. \quad (7)$$

В этой формуле символом G обозначен генератор текстов, принимающий значения G_i , из множества различных моделей его работы; T^{gen} – текст, сгенерированный посредством G .

Простейшая модель работы генератора текстов G_0 представляется дискретным стационарным процессом с независимыми значениями и равновероятными состояниями. Следующая модель – G_1 представляет собой дискретный стационарный процесс с независимыми значениями и состояниями, которые имеют вероятности, описанные в T_{zn}^1 . На следующих этапах генераторы текстов базируются на модели цепи Маркова [5]. Работа генератора G_2 моделируется односвязной цепью Маркова с вероятностями перехода T_{mark}^1 . Пример таким образом экспериментально сгенерированного текста:

“ПЕНИВИДЕНОВЖЕВСТ ВАПЙТЬОМИТРЦЯ ЗАСІТОМ’Ю Д ПЕМІЙ ДЖКОМЕЙ А ПОДИРДКИ ГОБРОГА”.

Аналогично, генератор G_3 использует модель двусвязной цепи Маркова с вероятностями перехода T_{mark}^2 , а генератор G_4 – модель трехсвязной цепи Маркова с вероятностями перехода T_{mark}^3 .

Текст:

“ВЖДЕНЕХИТИТТЯМ ВУ МАТНО ВИВ ЯКУ МОВАВРИЖА ВОПШКАЛА ЛЮБКАЗ СТУТЙСЬ ДОРЧУ АРИЗНА”

был экспериментально получен в результате работы G_3 , а текст:

“ДУМАЛИ ДЯКУСЬ НУТИ НЕ ПОДИНИ СЬОГОМУ КИНУ ХАЙ НЕІ ТУТ СТАНА ПРИТИ У ВИЛИПА”

был сгенерирован с помощью G_4 .

Как видно, последний текст наиболее близок к украинскому тексту. Таким образом, чем выше связность цепи Маркова, используемой в модели работы генератора текстов, тем ближе к реальному генерируемый им текст.

Используя статистические методы можно проводить исследования текста на предмет того, каким языком (или какими языками) написан текст,

Таблица 4

Характеристика работы генератора текстов (на 1000 символов)

Модель работы генератора текстов	Общее количество «слов»	Средняя длина «слова»	Средняя длина реальных слов	Количество реальных слов	% реальных слов
Дискретный стационарный случайный процесс с независимыми значениями и равновероятными состояниями	29	33,51	0	0	0
Дискретный стационарный случайный процесс с независимыми значениями, у которого распределение состояний совпадает с распределением символов текста	137	6,3	1,25	4	2,92
Однородная односвязная цепь Маркова	116	7,62	2,36	11	9,48
Однородная двусвязная цепь Маркова	119	7,39	2,77	26	21,85
Однородная трехсвязная цепь Маркова	133	6,52	5,05	65	48,87

формальное сравнение текстов с целью выявления плагиата, сравнение стилей (определение авторства), определение темы текста и многое другое. Вид статистического портрета можно выбрать исходя из задач исследования. Например, для исследования фонетической системы не обязательно строить статистический портрет лексической или грамматической систем. Далее приведем примеры наших подходов к решению конкретных задач.

3. Анализ студенческих работ на плагиат

Этот анализ является одним из относительно простых видов анализа текста. Мы полагаем, что при списывании студент не изменяет текст исходной работы, поскольку с одной стороны, сложность редактирования можно сравнить со сложностью написания новой работы, а с другой — если студент отредактировал текст работы, то этот новый текст можно уже рассматривать как новую работу, так как студент не только внимательно ее прочитал, но и понял, и, возможно, высказал некоторые свои мысли.

Для проведения анализа была создана тематически сгруппированная база данных студенческих работ. Текст тестируемой работы сравнивался со всеми работами из базы по данной тематике.

Статистический портрет в данном исследовании состоит из распределения слов и их комбинаций длиной до 3. Пример анализа студенческой работы на плагиат изображен на рисунке ниже:

На рис. 1 в левом окне — текст тестируемой студенческой работы, в правом — текст реферата из базы данных, с которым обнаружено сходство, совпадающие части выделены синим цветом. Программа автоматически проводит анализ текста реферата, результатом которого является вывод о том написана ли работа самостоятельно или списана с другой работы; в случае обнаружения плагиата, программа выдает список работ из базы, с которыми обнаружено сходство, для того чтобы преподаватель смог увидеть степень сходства текстов и показать, из какого источника работа списана. Существует возможность настройки точности работы программы.

4. Сравнение предвыборных программ политических партий и блоков

Статистический портрет для этого анализа включает в себя распределение слов и их комбинаций длиной до 3, основным компонентом является распределение слов по частоте. Для анализа были выбраны предвыборные программы политических

Рис. 1. Результат анализа студенческих работ

партий и блоков к парламентским выборам 2002 года с их официальных сайтов, в сравнении также участвовал текст Конституции Украины.

Программы, участвующие в исследовании, были пронумерованы. Для каждого текста программы политической партии были построены следующие множества: частотный список слов, встречающихся в тексте (первые 100 слов за исключением так называемых стоп-слов – часто употребляемых и служебных слов, не несущих смысловой нагрузки), все пары и тройки слов. Далее частотный список слов для каждой программы политической партии сравнивался с частотными списками слов всех остальных программ. В результате сравнения мы получили квадратную симметричную $k \times k$ матрицу P^1 , где k – количество программ партий, участвующих в исследовании. Каждый элемент этой матрицы $p^1[i, j]$ соответствует проценту совпадающих слов в программах с номерами i и j .

$$p^1[i, j] = p^1[j, i], p^1[i, i] = 1. \quad (8)$$

Аналогичным образом сравнивались множества пар и троек слов для каждой программы партии со всеми остальными программами. В результате сравнения мы получили квадратные $k \times k$ матрицы P^2 , P^3 , где k – количество программ партий, участвующих в исследовании. Каждый элемент матрицы P^2

$$p^2[i, j] = (N^2(i, j)/N^2(i)) * 100\%, \quad (9)$$

где $N^2(i, j)$ – количество пар слов в программе партии с номером i , совпадающих с парами слов партии j , $N^2(i)$ – общее количество пар слов в программе партии с номером i .

$$p^2[i, j] \neq p^2[j, i], p^2[i, i] = 1. \quad (10)$$

Аналогично для матрицы P^3 :

$$p^3[i, j] = (N^3(i, j)/N^3(i)) * 100\%, \quad (11)$$

$$p^3[i, j] \neq p^3[j, i], p^3[i, i] = 1. \quad (12)$$

Степень сходства программ партий с номерами i и j мы определяем таким образом:

$$p(i, j) = (p^1[i, j] + 2p^2[i, j] + 3p^3[i, j])/6. \quad (13)$$

В результате получаем $k \times k$ матрицу P , которая является результатом сравнения программ партий:

$$P = (P^1 + 2P^2 + 2P^3)/6. \quad (14)$$

Программы политических партий, участвующих в исследовании, были пронумерованы согласно списку:

1. Аграрная партия
2. Народно-демократическая партия
3. Конституция Украины
4. Партия регионов
5. Партия промышленников и предпринимателей
6. Народный Рух Украины
7. Социалистическая партия
8. Трудовая Украина
9. СДПУ(О)
10. УРП Собор

11. БЮТ

12. Наша Украина

13. Коммунистическая партия

В приведенных ниже матрицах элементы округлены до целой части.

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	100	48	34	45	52	55	53	46	64	35	28	30	40
2	48	100	29	42	46	51	45	41	61	41	37	39	44
3	34	29	100	27	27	31	29	25	36	33	37	34	38
4	45	42	27	100	45	44	42	39	56	41	38	41	44
5	52	46	27	45	100	43	44	38	53	44	46	46	49
6	55	51	31	44	43	100	31	41	42	51	47	48	54
7	53	45	29	42	44	31	100	38	45	44	44	44	46
8	46	41	25	39	38	41	38	100	52	41	41	44	46
9	64	61	36	56	53	42	45	52	100	58	57	58	63
10	35	41	33	41	44	51	44	41	58	100	34	30	35
11	28	37	37	38	46	47	44	41	57	34	100	34	19
12	30	39	34	41	46	48	44	44	58	30	34	100	31
13	40	44	38	44	49	54	59	46	63	35	19	31	100

Рис. 2. Матрица P^1

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	100	12	7	9	11	13	11	10	17	6	1	4	2
2	6	100	5	9	10	11	7	8	15	6	1	4	0
3	2	3	100	3	3	5	4	3	7	2	0	2	0
4	4	8	5	100	10	9	7	7	13	5	1	4	1
5	4	7	4	7	100	8	7	6	13	4	1	3	1
6	3	6	4	5	6	100	5	5	10	3	0	3	0
7	3	4	4	5	6	6	100	5	11	2	0	2	1
8	4	6	4	6	7	8	6	100	12	4	1	4	0
9	3	5	3	5	6	6	6	5	100	3	0	3	0
10	5	10	6	9	10	12	7	8	14	100	1	4	0
11	4	6	5	6	7	8	7	7	12	6	100	5	0
12	3	7	5	7	8	9	7	8	13	4	1	100	0
13	7	6	6	7	9	7	12	5	14	3	1	31	100

Рис. 3. Матрица P^2

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	100	3	1	1	1	2	2	1	2	0	0	0	0
2	1	100	0	1	1	2	0	1	2	1	0	0	0
3	0	0	100	0	0	1	0	0	0	0	0	0	0
4	0	1	0	100	1	1	0	0	2	0	0	0	0
5	0	1	0	1	100	1	0	0	2	0	0	0	0
6	0	1	0	0	0	100	0	0	1	0	0	0	0
7	0	0	0	0	0	0	100	0	1	0	0	0	0
8	0	0	0	0	1	1	0	100	1	0	0	0	0
9	0	0	0	0	1	0	0	0	100	0	0	0	0
10	0	1	1	1	1	2	0	1	1	100	0	0	0
11	0	0	0	0	0	0	0	1	1	0	100	0	0
12	0	0	0	0	1	1	0	1	2	0	0	100	0
13	1	0	0	0	1	0	2	0	1	0	0	0	100

Рис. 4. Матрица P^3

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	100	14	9	11	13	15	14	12	17	8	5	6	7
2	11	100	7	11	12	13	10	10	16	9	7	8	7
3	6	6	100	6	6	7	6	5	8	6	6	6	6
4	9	10	6	100	11	11	9	9	15	9	7	8	8
5	10	11	6	10	100	10	10	8	14	9	8	9	9
6	10	11	7	9	9	100	7	9	11	10	8	9	9
7	10	9	6	9	9	7	100	8	12	8	7	8	8
8	9	9	6	9	9	10	8	100	13	8	7	9	8
9	12	12	7	11	11	9	10	10	100	11	10	11	11
10	8	11	8	10	11	14	10	10	15	100	6	6	6
11	6	8	8	8	8	11	10	10	14	8	100	7	3
12	6	9	7	9	11	12	10	11	15	6	6	100	5
13	10	9	8	10	12	11	15	9	16	7	4	6	100

Рис. 5. Матрица P

5. Сравнение текстов словарей

В последнее время в связи со значительной активизацией лексикографической деятельности, появлением на рынке словарной продукции, имеющей явно выраженные признаки «клонов» известных словарей, возникает необходимость установить степень отличия между текстами такого рода изданий. Для решения таких задач также успешно применяются статистические методы. При этом возможно сравнение как по отдельным словарным статьям, так и в целом по всему корпусу словаря.

В Украинском языково-информационном фонде НАН Украины выпускается серия словарей «Словники України». В рамках этой серии – ежегодно издается орфографический словарь украинского языка. Составляется новый большой (в 20 томах) толковый «Словник української мови» (СУМ), прототипом которого является изданный в 70-80 годы прошлого столетия 11-томный «Словник української мови». Для ответа на вопрос, насколько отличаются тексты словарей разных изданий, был разработан инструментарий, позволяющий исследовать тексты словарей на предмет сходства и отличия. В основе методики сравнения лежит сравнение множеств цепочек символов, ограниченных пробелами. При исследовании текстов СУМа проводилось сравнение троек словоформ, входящих в 11- и 20-томник. Ниже приве-

дены фрагменты словарных статей с заголовочными словами «АНАРХІЧНО» – «АНАТОМ» из 11- и 20-томника и результат их сравнения, произведенный составленной нами программой. В левом окне – текст 11-томного СУМ, в правом – 20-томного (красным цветом выделены совпадающие части).

В результате анализа выяснилось, что текст первого тома 20-томного СУМа совпадает с соответствующим текстом 11-томника примерно на 37% согласно разработанной нами методике. Таким образом, можно вполне определенно утверждать, что 20-томник – совершенно новый лексикографический труд.

Выводы

Имея лишь текст на естественном языке, с помощью статистических методов, используя только формальные математические модели, можно проводить разнообразные исследования текстов. Процесс исследования текстов целесообразно разбить на три стадии: выбор параметров статистического портрета, статистическая обработка текста, в результате которой получаем статистический портрет с заданными параметрами и непосредственное исследование текстов через их статистические портреты. Получение лексических портретов текстов позволяет, абстрагировавшись от содержания текста, проводить дальнейшие исследования

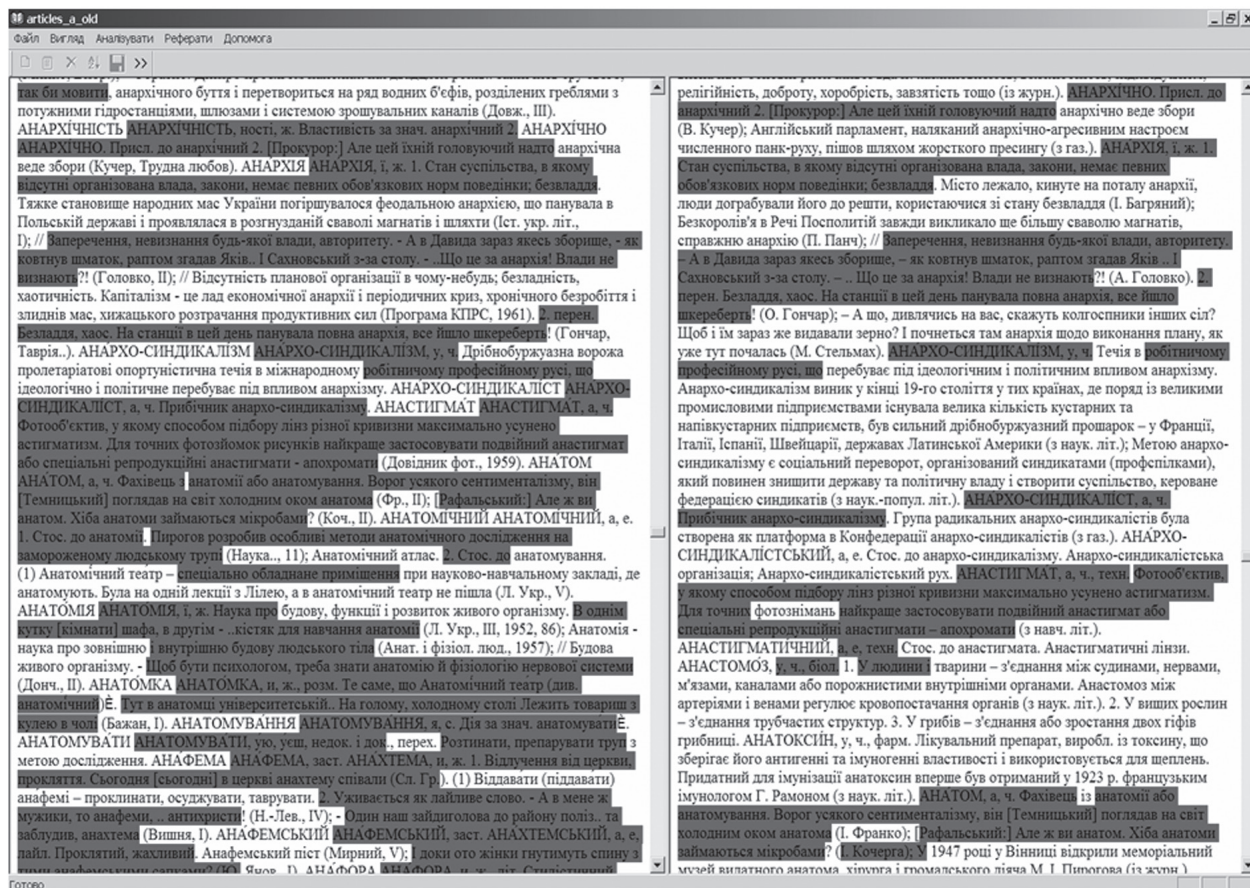


Рис. 6. Сравнение фрагментов 11-томного и 20-томного СУМ

на множествах различных параметров. Параметры, входящие в статистический портрет, выбираются в зависимости от цели исследования. Полагаем, что статистические методы являются наиболее эффективными при проведении различного рода лингвистических экспертиз.

Список литературы: 1. Шевченко, И. В. Электронный грамматический словарь украинского языка [Текст] / И.В. Шевченко, А.Г. Рабулец, В.А. Широков // Труды международной конференции «Megaling'2005. Прикладная лингвистика в поиске новых путей». 27 июня – 2 июля 2005 года. Меганом, Крым, Украина. – 2005. – С. 124–129. 2. Словник української мови в 11 т. [Текст]. – К.: Наук. думка, 1970–1980. 3. Широков, В. А. Корпусна лінгвістика [Текст] / В.А. Широков, О.В.Бугаков, Т.О.Грязнухіна та ін. – К.: Довіра, 2005. – С. 126-169. 4. Рабулець, О. Г. Лінгвістичний корпус як середовище обробки та збереження інформаційних ресурсів [Текст] / О.Г. Рабулець, Н.М. Сидорчук // Прикладна лінгвістика та лінгвістичні технології. – К.: Довіра, 2008. – с. 316-328. 5. Вентцель, Е. С. Теория случайных процессов и ее инженерные приложения [Текст] / Е.С. Вентцель, Л. А. Овчаров. – М.: Наука, 1991. – С. 98-127.

Поступила в редколлегию 23.03.2010 г.

УДК 658.012.011.56

Природномовний текст як об'єкт статистичного аналізу / М. Ю. Крыгин // Біоніка інтелекту: наук.-техн. журнал. – 2010. – № 1 (72). – С. 75–82.

У статті описано статистичний підхід до розв'язання деяких задач, пов'язаних з обробкою природного тексту. Представлено теоретичне обґрунтування застосованого підходу, що базується на понятті статистичного портрета тексту. Відображено результати дослідження статистичних властивостей текстів, на які спирається наш підхід.

Табл. 4. Іл. 6. Бібліогр.: 5 найм.

UDK 658.012.011.56

Natural language text as an object of statistic analysis / M. Yu. Krygin // Bionics of Intelligence: Sci. Mag. – 2010. – № 1 (72). – P. 75–82.

In the article statistic approach to solving problems concerned with natural texts processing is described. Theoretical basis for used approach based on concept of statistic portrait of text is represented. Results of research of statistic properties of texts that underlie our approach are described.

Fig.: 6. Tab.: 4. Ref.: 5 items.

УДК 519.62



ПОБУДОВА ФУНКЦІОНАЛЬНИХ МОДЕЛЕЙ ЕЛЕМЕНТІВ КОМП'ЮТЕРНОЇ ЛІНГВІСТИКИ

І.Д. Вечірська¹, Т.М. Федорова², Г.Г. Четвериков³

¹ ХНУРЕ м. Харків, Україна, ira_se@list.ru

² ХНУРЕ м. Харків, Україна, tanja_fedorova@mail.ru

³ ХНУРЕ м. Харків, Україна, chtvergg@kture.kharkov.ua

Стаття присвячена аналізу структури комп'ютерної лінгвістики та побудові функціональних моделей для опису задач та процесів обробки мовної інформації в корпусній лінгвістиці, машинному перекладі та системах пошуку та класифікації. Наведені приклади автоматичного розпізнавання символів, автоматичного реферування текстів, створення електронних словників, тезаурусів, онтологій, автоматичного розпізнавання мови і синтезу мовлення, оптичного розпізнавання символів.

ЛІНГВІСТИКА, КОМП'ЮТЕРНА ЛІНГВІСТИКА, КОРПУС, МАШИННИЙ ПЕРЕКЛАД, ІН-
ФОРМАЦІЙНО-ПОШУКОВА СИСТЕМА, СЛОВНИК, МОДУЛЬ, АНАЛІЗ

Вступ

Мова є найважливішим засобом комунікації в суспільстві і тісно пов'язана з мисленням і свідомістю. Мовознавство являється однією з центральних наук і входить до кола гуманітарних наукових дисциплін, які досліджують людину і людське суспільство. Лінгвістика вивчає не тільки існуючі (існуючі або можливі в майбутньому) мови, а й людську мову взагалі. Лінгвістику можна розділити на наукову (яка передбачає побудову лінгвістичних теорій) і практичну. Найчастіше під лінгвістикою мають на увазі саме наукову лінгвістику [1].

Лінгвістика безпосередньо вивчає лише факти мови або мовні явища, тобто мовні акти носіїв живої мови разом з їх результатами (текстами) або мовний матеріал (обмежена кількість письмових текстів на мертвій мові, якою вже ніхто не користується як засобом спілкування).

Лінгвістика займається не тільки спостереженням, реєстрацією та описом фактів мови, а й висуненням гіпотез для пояснення цих фактів, формулюванням гіпотез у вигляді теорій і моделей, що описують мову, їх експериментальною перевіркою і спростуванням; прогнозуванням мовної поведінки. Пояснення фактів буває внутрішнім (через мовні ж факти) або зовнішнім (через факти фізіологічні, психологічні, логічні або соціальні).

Лінгвістику в широкому сенсі слова (пізнання мови і передача результатів цього пізнання іншим людям) можна розділити на наступні напрямки:

- теоретична лінгвістика (наукова) передбачає побудову лінгвістичних теорій, розглядає всі аспекти і проблеми, пов'язані з мовою, склад і вживання, загальні закономірності пристосування та розвиток мови;

- прикладна лінгвістика спеціалізується на вирішенні практичних завдань, пов'язаних з вивченням мови, а також на практичному використанні лінгвістичної теорії в інших областях. Наприклад методика вивчення рідної та нерідної мови, лексикографія,

переклад, дешифрування, орфографія, транслітерація, розробка термінології і тому подібне;

- практична лінгвістика є сферою, де реально проводяться лінгвістичні експерименти, що мають на меті верифікацію положень теоретичної лінгвістики та перевірку ефективності продуктів, що створюються прикладною лінгвістикою. Наприклад навчання дітей рідній мові, вивчення іноземної мови, переклад, викладання рідної та іноземної мови, літературне редагування, коректура, практична логопедія, побутова і художня словотворчість, мовна політика, створення нових писемностей і навчання грамоті, і тому подібне) [2].

В прикладній лінгвістиці можна виділити наступні напрямки, пов'язані з вивченням мови:

- лексикографія – теорія та практика складання словників;
- лінгводидактика – наука про розробки методик навчання іноземній мові;
- термінознавство – наука про впорядкування та стандартизацію науково-технічної термінології;
- перекладознавство – теорія перекладу.

Основними напрямками прикладної лінгвістики (англ. applied linguistics), які пов'язані з практичними впровадженнями, є наступні:

- Комп'ютерна лінгвістика (англ. computational linguistics):
 - корпусна лінгвістика (створення та використання електронних корпусів текстів);
 - машинний переклад;
 - розробка систем пошуку та класифікації;
 - автоматичне розпізнавання символів (системи автоматичної корекції правопису, такі як автокоректори, спелчекери і так далі);
 - автоматичне реферування текстів (системи авто реферування, наприклад Automatic Text Summarization);
 - створення електронних словників, тезаурусів, онтологій (електронні одномовні та перекладні словники загального вживання: тлумачні,

орфографічні, термінологічні, тощо, наприклад LINGVO фірми АBBYY; КОНТЕКСТ фірми ІН-ФОРМАТИК; УЛІС фірми ПроЛінг);

– автоматичне розпізнавання мови (системи акустичного розпізнавання – Automatic Speech Recognition, і синтезу мовлення – Text-To-Speech Engine);

– оптичне розпізнавання символів (системи оптичного розпізнавання, наприклад OCR – Optical Character Recognition, система OCR Fine Reader фірми АBBYY);

– логічний аналіз текстів (системи логічного аналізу змісту текстів та локалізації знань, наприклад впровадження фірмою Text Analysis International Inc. наприкінці 2001 року інтегровано системи розробки Visual Text™ для побудови програмного забезпечення глибокого аналізу тексту).

• Лінгвістична експертиза. Один з видів лінгвістичного дослідження, який призначається уповноваженою особою (органом) з метою встановлення юридично значущих фактів.

У лінгвістичному аспекті лінгвістична експертиза – це вид дослідження об'єктів, що встановлює істинність / хибність або можливість / неможливість висловлювань опису об'єкта (об'єктів). Об'єктом лінгвістичної експертизи є продукти мовленнєвої діяльності (висловлювання, тексти, лексеми, словесні позначення товарних знаків, тощо). Предметом експертизи є певний лінгвістичний зріз досліджуваного об'єкта.

Основні задачі, які вирішує лінгвістична експертиза наступні:

– виявлення змісту слів, фраз, пропозицій, що містяться у суперечливих текстах (на електронних носіях та в мережі Інтернет, офіційних і приватних документах, тощо) і встановлення форми їх вираження; встановлення змісту поняття, що виражається словом, словосполученням, пропозицією; встановлення ступеня адекватності інтерпретації одного тексту іншим текстом (визначення змісту) та інше;

– оцінка тексту з точки зору вирішення питання про його автора; визначення авторських, суперечливих, анонімних текстів, різновиди літературного плагіату та інше;

– встановлення схожості товарних знаків; встановлення наявності в найменуваннях прихованої та явної пропаганди (наркотиків, насильства, міжнародної ворожнечі, тощо), прихованої реклами і антиреклами та інше.

• Наука про впорядкування та стандартизацію науково-технічної термінології. Об'єктом упорядкування у термінознавстві є термінологія, тобто природно сформована сукупність термінів певної галузі знання або її фрагменту. Термінологія піддається систематизації, потім аналізу з метою виявлення її недоліків та методів їх усунення і, нарешті,

нормалізації. Результат цієї роботи представляється у вигляді терміносистеми – упорядкованої множини термінів із зафіксованими відносинами між ними, що відображають відносини між поняттями, що називають ці терміни.

Таким чином, задача систематизації знань про лінгвістику як науку для спрощення розуміння етапів обробки лінгвістичної інформації та її подальшої реалізації за допомогою обчислювальної техніки є актуальною на сьогоднішній день.

Метою даної роботи є представлення комп'ютерної лінгвістики у вигляді ієрархічної структури та побудова функціональних моделей її елементів.

1. Побудова структури комп'ютерної лінгвістики

Комп'ютерна лінгвістика – це напрямок у прикладній лінгвістиці, який вивчає застосування математичних моделей для опису лінгвістичних закономірностей. За способом дослідження лінгвістичної інформації комп'ютерна лінгвістика поділяється на дві великі частини:

• вивчення способів застосування обчислювальної техніки в лінгвістичних дослідженнях (застосування відомих математичних методів, наприклад статистичної обробки, для виявлення лінгвістичних закономірностей);

• осмислення текстів, написаних природною мовою (створення математичних моделей для розв'язання лінгвістичних завдань та розробка програм, які функціонують на основі цих моделей). Друга частина комп'ютерної лінгвістики тісно пов'язана з розділом штучного інтелекту, що займається розробкою систем опрацювання природної мови (йдеться про засоби обробки як текстової інформації, так і природного мовлення).

У вузькому сенсі проблематику комп'ютерної лінгвістики часто пов'язують з міждисциплінарним прикладним напрямком «обробка природної мови», тобто розробка методів, технологій і конкретних систем, які забезпечують спілкування людини з ЕОМ природною або обмежено природною мовою.

На рис. 1 схематично представлена структура комп'ютерної лінгвістики. Далі розглянемо більш детально кожний з підрозділів комп'ютерної лінгвістики.

2. Корпусна лінгвістика

Корпусна лінгвістика – розділ комп'ютерної лінгвістики, у якому дослідження проводяться за допомогою комп'ютерних лінгвістичних корпусів. Під лінгвістичним корпусом текстів розуміємо великий, уніфікований, структурований, розмічений, філологічно компетентний масив мовних даних, представлений в електронному вигляді й призначений для розв'язання різних лінгвістичних завдань [3].

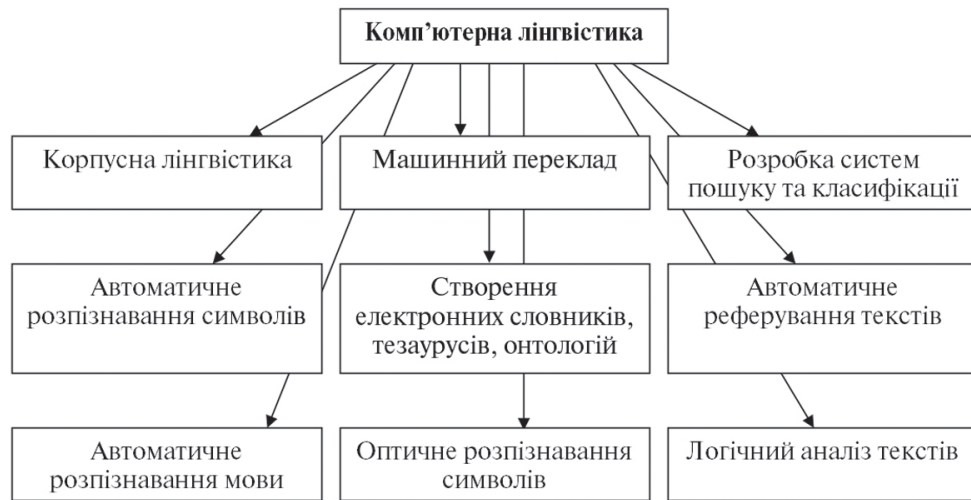


Рис. 1. Структура задач комп'ютерної лінгвістики

Першим великим комп'ютерним корпусом вважається Брауновський корпус, створений у 1960-х роках в Університеті Брауна, що містив 500 фрагментів текстів по 2 тисячі слів у кожному, які були опубліковані англійською мовою в США в 1961 році. У результаті він став стандартом в 1 млн. слововживань для створення представницьких корпусів на інших мовах. У зв'язку із зростанням комп'ютерних потужностей, здатних працювати з великими обсягами текстів, у 1980-і роки в світі було зроблено кілька спроб створити корпуси більшого розміру. У Великобританії такими проектами були Банк Англійської Мови (Bank of English) і Британський Національний Корпус (British National Corpus, BNC). В Росії таким проектом був Машинний Фонд російської мови, який було створено за ініціативою А. П. Єршова. В Україні таким проектом є репрезентативний лінгвістичний кор-

пус Українського мовно-інформаційного фонду НАН України, який функціонує в промисловому режимі приблизно з 2000 року і налічує понад 25 млн. слововживань [3, 4].

Найважливішим при створенні лінгвістичних корпусів є поняття розмітки, відмежовування корпусів текстів від простого зібрання електронних текстів. Розмітка полягає в тому, що текстам корпусу та його компонентам приписуються спеціальні мітки (індикатори) різних типів:

- зовнішні (зазвичай, це елементи бібліографічного опису: видання, рік, автор, тощо);
- структурні (описують структуру тексту: розділ, абзац, речення, тощо);
- лінгвістичні (лексикографічні та граматичні характеристики).

На рис. 2 представлено функціональну модель корпусної лінгвістики.

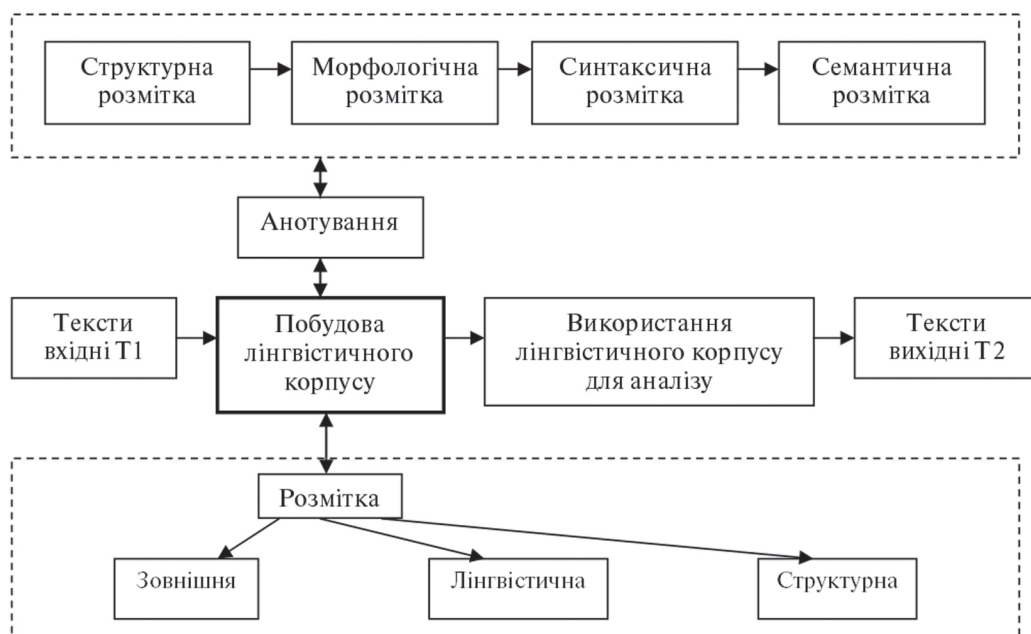


Рис. 2. Функціональна модель побудови лінгвістичних корпусів

3. Машинний переклад

Під час машинного перекладу (МП) можна зустрітись з такими проблемами:

- проблема створення великих словників для систем;
- проблема навчання системи розпізнавання стійких виразів;
- проблема запису усіх правил побудови речення і перекладу у вигляді програми.

Одна із систем, яка намагається вирішити ці проблеми, розробляється в Українському мовно-інформаційному фонді. В цій системі використовуються принципи індуктивно-інтерактивного моделювання процесу автоматичного перекладу з незалежними етапами аналізу вхідного тексту й синтезу вихідних текстів [5].

На рис. 3 представлена схема процесів машинного перекладу.

На першому етапі машинного перекладу відбувається аналіз вхідного тексту, який складається з чотирьох модулів аналізу:

- передморфологічний;
- морфологічний;
- контекстний;
- синтаксичний.

Кожен з цих модулів може використовуватися також і в інших системах обробки мовної інформації (автоматичне індексування, інформаційний пошук, автоматичне редагування, тощо). Сама ж

система аналізу є основою лінгвістичного забезпечення формального опису мови.

Алгоритми перекладу будуються на інформації про міжмовні відношення щодо лексичних, морфологічних та семантико-синтаксичних відповідностей між одиницями перекладу вхідного тексту і синтезованого вихідного. Регульовальну роль у цьому процесі відіграє граматики, що фіксує синтаксичні нормативні правила вихідної мови. Трансформаційні граматики (знаходження подібності та наявність у синтаксичній системі вихідних структур і правил їх перетворення (трансформації) за умови збереження незмінним лексичного складу вихідного змісту і синтаксичних відносин між лексемами) будуються на зіставленні грамастик обох мов.

Ядро системи машинного перекладу становить автоматичний багатомовний перекладний словник, який є компонентом інтегрованої лексикографічної бази Українського мовно-інформаційного фонду. Автоматичний багатомовний перекладний словник будується як відкрита неоднорідна, багатофункціональна лексикографічна база даних [6]. Словник інтегрує в собі лексикони мов, залучених до системи МП, забезпечує пошук перекладних еквівалентів з української мови на інші і навпаки. Автоматичний багатомовний перекладний словник формується на основі двомовних електронних словників із залученням комп'ютерних термінологічних баз даних, електронних тлумачних словників, словників синонімів, синтаксичних словників, що фіксують валентні властивості лексичних і граматичних мовних одиниць. Усі ці словники заносяться до загальної лексикографічної бази.

Автоматичний багатомовний перекладний словник інтегрується з граматичними словниками відповідних мов.

Граматичні словники використовуються як базовий інструмент на всіх етапах аналізу і синтезу МП.

Крім автоматичного багатомовного перекладного словника та граматичних словників до словникової бази включаються бінарні перекладні словники словосполучень як основний засіб представлення контекстного визначення багатозначних слів.

Для словників словосполучень використовується лінгвістична база з корпусів паралельних текстів для кожної пари мов, на які орієнтована система МП. Ця ж база є вихідним мовним матеріалом для розробки лінгвістичних алгоритмів аналізу, синтезу і трансформаційних грамастик, що лежать в основі алгоритмів перекладу.

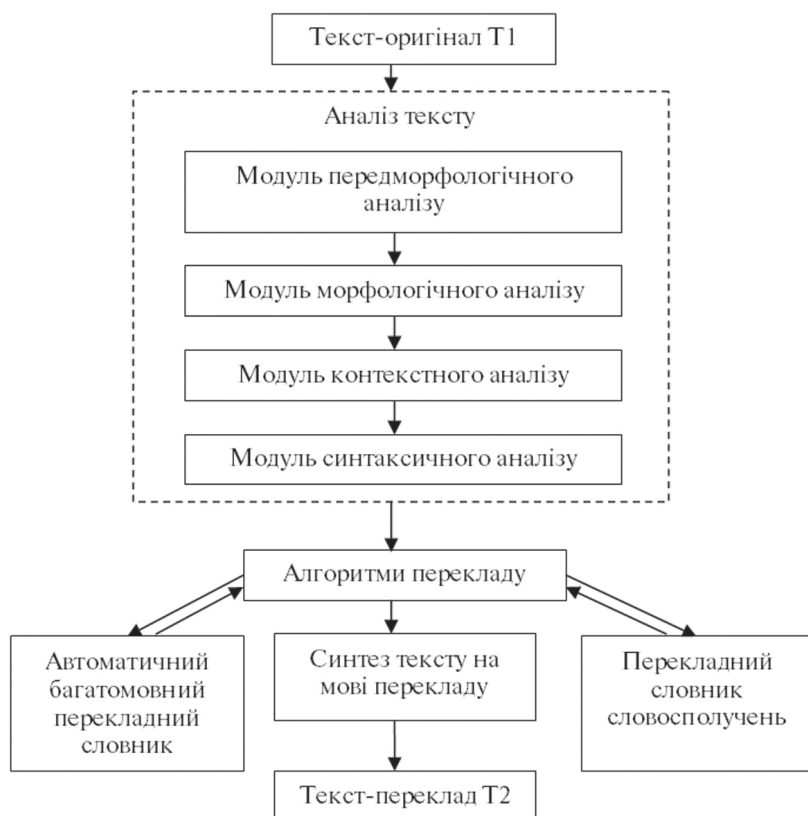


Рис. 3. Функціональна модель машинного перекладу

4. Системи пошуку та класифікації

Інформаційно-пошукова система виконує такі функції:

- зберігання великих обсягів інформації;
- швидкий пошук необхідної інформації;
- внесення, видалення і зміни інформації, що зберігається;
- подання вихідної інформації в зручному для людини вигляді.

На рис. 4 представлена загальна схема етапів пошуку та класифікації інформації.

Велике значення в сучасних повнотекстових інформаційно-пошукових системах приділяється морфологічному аналізу, тобто автоматичним засобам обробки окремих слів (як в текстах вихідних документів, так і в запитах користувача). Хоча слід зауважити, що не всі пошукові системи включають до своєї системи морфологічну обробку тексту (як приклад можна назвати систему AltaVista (<http://www.altavista.com/>). Недоліком цієї системи є слабе ранжування (ранжування визначає порядок видачі результатів пошуковим сервером: так, сторінки, на яких слова запиту зустрічаються частіше, отримують більш високий ранг і виводяться першими).

Використовуючи морфологічний аналіз в пошукових системах, розробники ставлять за мету покращення результату пошуку, а саме: збільшення повноти та точності пошуку. Коли відбувається пошук за канонічними формами слів, до результату пошуку потрапляють не тільки документи зі словом у тій формі, яка точно відповідає словоформі у запиті, але й інші документи, які містять різні форми даного слова (тобто обсяг інформації, що видається на запит користувача, справді збільшується). Ще однією перевагою використання канонічних індексних баз під час пошуку є спрощення інтерфейсу користувача. Найчастіше користувачу необхідно отримати на запит всі варіанти відповідей на запит. За відсутності автоматичного розширення слів його варіантами (коли пошук виконується не за однією, вказаною у запиті, словоформою, а за множиною усіх форм даного слова, що входять до його словозмінної парадигми) користувач зму-

шений вивчати і використовувати у своїх запитах формули або оператори відсікання.

Багато праць присвячено дослідженню впливу різних алгоритмів морфологічного аналізу на якість інформаційного пошуку в текстах на різних мовах. Результати цих досліджень дозволяють зробити висновок, що запровадження морфологічних модулів у пошукові системи дозволяє збільшити повноту і точність інформаційного пошуку [7]. Яскравим прикладом таких систем є програмні продукти серії Yandex (Мовний index).

Висновки

Нами було запропоновано та розроблено представлення комп'ютерної лінгвістики у вигляді ієрархічної структури та структурний опис задачі побудови функціональних моделей обробки мовної інформації на відповідних етапах.

Крім розглянутих вище підрозділів комп'ютерної лінгвістики, таких як корпусна лінгвістика, машинний переклад і системи пошуку та класифікації, є перспективи подальших досліджень процесів обробки інформації у таких напрямках, як:

- Автоматичне розпізнавання символів (системи автоматичної корекції правопису (автокоректори, спелчекери). Наприклад: ОРФО – система перевірки правопису, розробка фірми ІНФОРМАТИК; РУТА – система перевірки правопису; УЛІС – електронний українсько-російський та російсько-український словник; Ispell – GNU Project – Free Software Foundation, URL.
- Автоматичне реферування текстів (системи авто реферування. Наприклад: Inxight – комерційна система автоматичного реферування; Extractor – Модуль авто реферування; Text Analyst™ – програмний інструмент для аналізу змісту текстів, змістовного пошуку інформації, формування електронних архівів.
- Створення електронних словників, тезаурусів, онтологій (електронні одномовні та перекладні словники загального вжитку (глумачні, орфографічні, термінологічні тощо)). Наприклад: LINGVO фірми АВВУ, КОНТЕКСТ фірми ІНФОРМАТИК, УЛІС фірми ПроЛінг.

«АВВУ Lingvo 11 Багатомовна версія» – найбільш популярний багатомовний електронний словник, що містить 113 професійних словників і дозволяє перекладати з російської мови на іспанську, англійську, німецьку, французьку та італійську мови. Крім традиційного списку європейських мов, перекладач дає можливість працювати з нещодавно доданими китайською, турецькою, українською, португальською мовами, а також робить доступним словник афоризмів латинської мови.

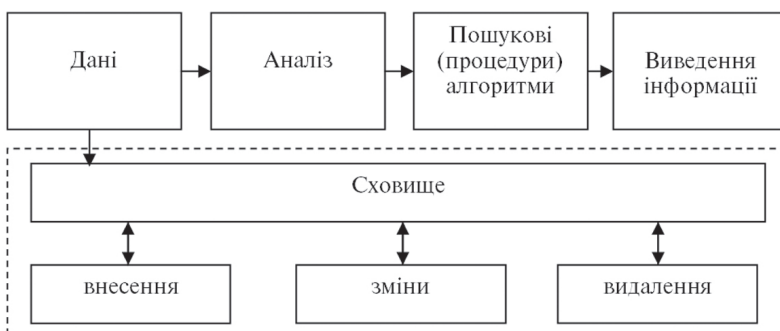


Рис. 4. Функціональна модель системи пошуку та класифікації

Електронні словники КОНТЕКСТ – це система професійних електронних словників. Містить великий набір багатомовних словникових баз як загальнолексичних, так і диференційованих по різних галузях знань. Активний розвиток електронного словника КОНТЕКСТ дозволило йому стати зручним і надійним помічником для тих, хто вивчає іноземні мови або працює з ними.

УЛІС – електронний російсько-український і українсько-російський словник. Основні можливості:

– в словникові статті включені докладні тлумачення значень слів, граматичні коментарі й приклади вживання слів у сталих словосполученнях;

– для багатозначних слів широко представлені варіанти перекладу;

– зручна пошукова система дозволяє легко і швидко знайти потрібне слово в будь-якій формі, його синоніми або близькі за значенням слова;

– додатково в словнику УЛІС реалізована можливість перекладу слова з будь-якої його форми з урахуванням особливостей морфології.

• Автоматичне розпізнавання мови (системи акустичного розпізнавання) і синтезу мовлення. Наприклад, <http://www.speech.com.ua> – сайт з розпізнавання та синтезу мовлення в Україні; Voice Type Dictation, Voice Pilot и ViaVoice от IBM : <http://www.ibm.com/software/speech>.

• Оптичне розпізнавання символів (системи оптичного розпізнавання). Наприклад система OCR Fine Reader фірми ABBYY.

Більшість програм оптичного розпізнавання тексту (OCR Optical Character Recognition) працюють з растровим зображенням, яке отримано через факс-модем, сканер, цифрову фотокамеру або інший пристрій.

Система ABBYY FineReader призначена для конвертації в редаговані формати відсканованих документів, PDF-документів та файлів зображень, включаючи цифрові фотографії.

Список літератури: 1. Четвериков, Г.Г. Концептуально-методологічний підхід до моделювання природної мови алгебро-логічними засобами [Текст] / І.Д. Вечірська, Т.Н. Федорова та ін. // Тезиси доклади Международной

научной конференции “Горизонты прикладной лингвистики и лингвистических технологий” (MegaLing’2009). – 20-27 сентября 2009, Украина, Киев. – С. 68. 2. *Лингвистика* [Электронный ресурс] / Режим доступа : [www/URL: http://ru.wikipedia.org/w/](http://ru.wikipedia.org/w/) – 24.02.2010 г. – Загл. с экрана. 3. Широков, В.А. Корпусна лінгвістика [Текст] / В.А. Широков, О.В. Бугаков, Т.О. Грязнухіна та ін. ; відповідальний редактор В.А. Широков. – К. : „Довіра”, 2005. – 472 с. 4. *Корпусная лингвистика* [Электронный ресурс] / Режим доступа : [www/URL: http://ru.wikipedia.org/wiki/](http://ru.wikipedia.org/wiki/) – 11.02.2010 г. – Загл. с экрана. 5. Широков, В.А. Очерк основных принципов квантовой лингвистики [Текст] / В. А. Широков // Бионика интеллекта. – 2007. – № 1 (66) – С. 25 – 32. 6. Грязнухіна, Т.О. Автоматичний багатомовний перекладний словник [Текст] / Т.О. Грязнухіна, Т.П. Любченко – Вісник лінгвістичного університету, том 6. – № 2. – Київ, 2003. – С. 68-71. (Особистий внесок: організація даних та структура АБПС). 7. Губин, М.В. Влияние морфологического анализа на качество информационного поиска [Текст] / М.В. Губин, А.Б. Морозов : Труды Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» RCDL2006 URL: http://www.rcdl2006.uniyar.ac.ru/papers/paper_67_v2.pdf.

Надійшла до редколегії 25.03.2010 р.

УДК 519.62

Построение функциональных моделей элементов компьютерной лингвистики / И.Д. Вечирская, Т.Н. Федорова, Г.Г. Четвериков // Бионика интеллекта: науч.-техн. журнал. – 2010. – №1 (72). – С. 83–88.

В статье было предложено и разработано представление компьютерной лингвистики в виде иерархической структуры и структурное описание задачи построения функциональных моделей обработки языковой информации на соответствующих этапах.

Л. 4. Библиогр.: 7 назв.

UDK 519.62

Construction functional models of computer linguistics elements / I.D. Vechirska, T.N. Fedorova, G.G. Chetverikov // Bionics of Intelligence: Sci. Mag. – 2010. – № 1 (72). – P. 83–88.

In article it was offered and representation the computer linguistics in the form of hierarchical structure and the structural description problem of construction functional models processing the language information at corresponding stages is developed.

Fig. 4. Ref.: 7 items.

УДК 681.3.07



ИСПОЛЬЗОВАНИЕ УРОВНЕЙ ВЕЙВЛЕТ-ПРЕОБРАЗОВАНИЯ ДЛЯ ВЫДЕЛЕНИЯ КОНТУРА ИЗОБРАЖЕНИЯ

В. А. Любченко¹, О. А. Кобылин²^{1,2} ХНУРЭ, г. Харьков, Украина, lubchenko_v@mail.ru

Использование уменьшенных копий изображения, полученных в результате применения вейвлет-преобразования, позволяет эффективно выделять контур изображения в системах технического зрения. Рассмотрено применение основных методов выделения контура на различных уровнях вейвлет-преобразования изображения. Использование уменьшенных копий изображения позволяет сократить время, необходимое для выделения контура изображения, а также сократить время нормализации.

КОНТУР, ВЕЙВЛЕТ-АНАЛИЗ, СИСТЕМА ТЕХНИЧЕСКОГО ЗРЕНИЯ, ЭТАЛОН

Введение

Обработка информации всегда играла важную роль в развитии научно технического прогресса. Количество визуальной информации, которое необходимо обрабатывать в реальном масштабе времени, постоянно увеличивается. Поэтому необходимо разрабатывать новые методы обработки изображений для систем технического зрения.

Среди подобных задач важное место занимают задачи, связанные с обработкой изображений, подверженных геометрическим искажениям [1].

В настоящее время в системах технического зрения актуальной и нерешенной в полном объеме является выделение границ изображения для последующего распознавания объектов, нормализации и установления их геометрических характеристик [1, 4]. Выделение границ на изображении можно рассматривать как отдельную задачу, не связанную с сегментацией. Однако к эффективным методам сегментации относится выделение границ объектов. Граничными точками считают в таком случае точки резкого перепада функции яркости. Для нахождения границ используется, как правило, численное дифференцирование. Одним из распространенных методов является градиентный метод, известный также как метод контрастирования или пространственного дифференцирования. Для этого на практике используют модуль градиента функции яркости

$$|\Delta f(x, y)| = \sqrt{\left(\frac{df(x, y)}{dx}\right)^2 + \left(\frac{df(x, y)}{dy}\right)^2},$$

который пропорционален изменению функции яркости в данной точке и не зависит от направления контура. Применяя фильтр к изображению, получают изображение градиентов. Это преобразованное изображение отличается от исходного ярко выраженными перепадами яркости. Точка на изображении является контуром, если яркость изображения градиентов превышает заданный порог [7].

Как правило, вейвлет анализ применяется для анализа одномерных сигналов. В настоящее время вейвлет-преобразование активно применяют для

обработки практически всех основных графических объектов: кривых, поверхностей, сплошных трехмерных тел. Также вейвлет-анализ находит применение и в задачах компьютерного зрения, распознавания и классификации образов.

1. Вейвлет-преобразование изображения и выделение контура

Цель работы состоит в применении вейвлет-преобразования для выделения контура объекта.

Объектом исследования в системах компьютерного зрения (СКЗ) является трехмерный объект O , не изменяющий своих абсолютных размеров в евклидовом пространстве.

Одним из известных методов, работающих в рамках геометрических преобразований, является нормализация [1]. Процедура нормализации заключается в автоматическом определении неизвестных параметров преобразования, которому подвергнуто входное изображение, и последующем приведении его к эталонному виду. В процессе нормализации изображение замещается эквивалентным ему. Преобразования осуществляются с помощью операторов нормализации F , называемых нормализаторами, а вычисление параметров выполняется функционалами Φ , действующими на множестве изображений. Существуют параллельные, последовательные, параметрические и следящие нормализаторы. Однако известные нормализаторы эффективно работают только для базовых преобразований, таких как смещения, повороты, растяжения, косые сдвиги.

Под вейвлет-преобразованием понимается разложение сигнала по системе вейвлетов [2-3]. Данное преобразование позволяет получить хорошее приближение изображения. Для описания контура изображения используем вейвлеты Хаара.

Любую функцию $f \in L^2(R)$ можно представить в виде ряда:

$$f(x) = \sum_{i=-\infty}^{\infty} c_i \psi_i(x).$$

Используя ортогональный вейвлет Хаара, определяемый соотношением

$$\psi_i^j(x) = \psi(2^j x - i), \text{ где } i = 0, \dots, 2^j - 1,$$

причем

$$\psi(x) = \begin{cases} 1, & x \in [0; 0,5) \\ -1, & x \in [0,5; 1), \\ 0, & x \notin [0; 1) \end{cases}$$

произведем непрерывное прямое вейвлет преобразование, путем вычисления вейвлет коэффициентов по формуле:

$$c_i = \langle f(x), \psi(a, b, t) \rangle = |a|^{-1/2} \int_{-\infty}^{\infty} f(x) \psi\left(\frac{t-b}{a}\right) dt,$$

где a – масштабный коэффициент; b – параметр сдвига.

Таким образом, каждая функция $f \in L^2(R)$ может быть представлена в виде набора числовых коэффициентов. Размер (величина) коэффициентов зависит от масштаба и параметров сдвига базисного вейвлета (Вейвлет Хаара). Полученные коэффициенты обеспечивают информацию, не зависящую от разрешения изображения.

Обратное непрерывное вейвлет преобразование можно осуществить по формуле:

$$f(x) = C_{\psi}^{-1} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} c_i \psi\left(\frac{t-b}{a}\right) \frac{dadb}{a^2},$$

где a – масштабный коэффициент; b – параметр сдвига; C_{ψ}^{-1} – нормализующий коэффициент; c_i – коэффициенты разложения.

Произведем вейвлет-преобразование изображения $B(x, y) \in L_2(R)$, имеющего размер 256×256 (рис. 1) как банк фильтров, применив следующие формулы:

$$vv_j(k, l) = \frac{1}{2} \sum_{n=0}^{2^j-1} \sum_{m=0}^{2^j-1} h'(2k-n) h'(2l-m) vv_{j-1}(n, m),$$

$$wv_j(k, l) = \frac{1}{2} \sum_{n=0}^{2^j-1} \sum_{m=0}^{2^j-1} g'(2k-n) h'(2l-m) vv_{j-1}(n, m),$$

$$vw_j(k, l) = \frac{1}{2} \sum_{n=0}^{2^j-1} \sum_{m=0}^{2^j-1} h'(2k-n) g'(2l-m) vv_{j-1}(n, m),$$

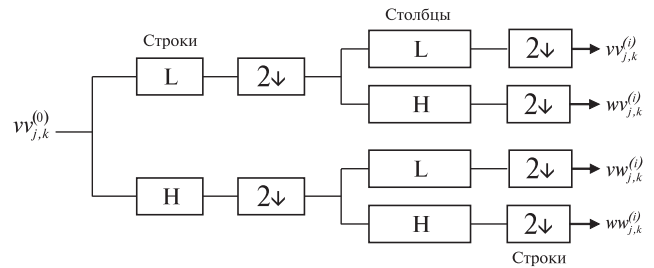
$$ww_j(k, l) = \frac{1}{2} \sum_{n=0}^{2^j-1} \sum_{m=0}^{2^j-1} g'(2k-n) g'(2l-m) vv_{j-1}(n, m),$$

где j – степень разрешения; $m, n = 0 \dots 2^j - 1$, $k, l = 0 \dots 2^j - 1$; $j = 0 \dots J$, $L = 2^j - 1$, $h'(n) = h(L - n - 1)$; $g'(n) = g(L - n - 1)$; n и m – координаты пикселей изображения $vv_{j,k}^{(i)}$; k и l – координаты пикселей изображения $wv_{j,k}^{(i)}$, $vw_{j,k}^{(i)}$, $ww_{j,k}^{(i)}$; $h(t)$ и $g(t)$ – импульсные характеристики низкочастотного и высокочастотного фильтра.

Схема разложения и пример разложения тестового изображения представлен на рис. 2-3.



Рис. 1. Пример изображения



(L – низкочастотный фильтр, H – высокочастотный фильтр)

Рис. 2. Двухмерное дискретное ВП



Рис. 3. Результат разложения, 1-й уровень

В результате разложения получим уменьшенную копию с применением декомпозирующего фильтра h , исходного изображения $WB^{(0)}(x, y)$.

Рассмотрим задачу выделения контура изображения. Задача выделения контуров состоит в построении изображения границ объектов и выделения однородных областей. Контуром изображения считается некоторая линия, являющаяся границей перехода от объекта к фону шириной, как минимум, в один пиксель. Если на изображении присутствует шум, то возможно после обработки могут появиться ложные контуры, которые не являются границами объектов.

Произведем выделение контуров с использованием $WB^{(0)}(x, y)$ при помощи методов, основанных на применении масок размером 3×3 [6, 7]. Основными являются метод Робертса, Собела, Превита, Канни.

При обработке методом Собела новое значение каждого пикселя B' вычисляется следующим образом:

$$B' = \sqrt{X^2 + Y^2},$$

$$\text{где } X = (A_3 + 2 * A_4 + A_5) - (A_1 + 2 * A_8 + A_7),$$

$$Y = (A_1 + 2 * A_2 + A_3) - (A_7 + 2 * A_6 + A_5)$$

$$\text{матрицы } \begin{pmatrix} A_1 & A_2 & A_3 \\ A_8 & B & A_4 \\ A_7 & A_6 & A_5 \end{pmatrix}.$$

Метод Собеля позволяет избежать появления шума из-за сглаженной функции градиентов.

Определение границ реализованы при помощи математического пакета MatLab функцией edge().

Рассмотренные методы, основанные на применении масок 3×3 , дают практически одинаковые результаты. Недостатком этих методов является отсутствие автоматического выбора порога.

Качество обработки изображений определяется по подавлению фона и выделению объектов в виде связанных областей. Также существенную роль в определении контуров играет яркость объектов изображения, которая в реальных условиях зависит от многих факторов и может изменяться. Вследствие этого могут образовываться ложные края на изображении.

Набор уровней, полученный в результате вейвлет-преобразования изображения, позволяет проводить более детальный анализ исходного изображения. Переход к другим уровням разложения позволяет отойти от мелких и случайных деталей, лучше выявить «внутреннюю» структуру при анализе геометрических искажений.

В связи с этим возможно проводить выделение контура на разных уровнях разложения изображения. Изображения, полученные в результате вейвлет-разложения, лучше подходят для нормализации, чем исходное полутонное изображение.

Однако в результате вейвлет-преобразования на каждом шаге происходит изменение размера изображения в два раза (изображение пропускается через фильтры с полосой пропускания 0,5). Очевидно, что не каждый уровень разложения изображения подходит для операции нормализации. Поэтому необходимо определять оптимальный уровень вейвлет-разложения. Это особенно важно при проведении операций, связанных с геометрическими преобразованиями – нормализацией изображения.

Однако не каждый уровень разложения изображения подходит для операции выделения контура.

Это объясняется тем, что сигнал претерпевает значительные изменения сразу на многих масштабах исследования и, соответственно, характеризуется возрастанием вейвлет-коэффициентов для многих уровней детализации в то время, как на стационарных участках вейвлет-коэффициенты оказываются сгруппированными вблизи определенных масштабов. Выбор уровня разложения может быть сведен к отысканию моментов увеличения вейвлет-коэффициентов на значительном количестве уровней масштабирования.

Таким образом, нормализацию необходимо проводить на том уровне разложения, который максимально сохраняет структуру изображения.

Первый уровень разложения является наиболее информативным.

В этом случае нормализацию для различных моделей, с использованием полученного контура, геометрических преобразований можно выполнить, используя следующие формулы:

Для группы $G_{c,x}$ –

$$WF_C(B) = WB(x + \Phi w^i_1, y),$$

где

$$\Phi w^i_1 = \frac{\sum_{n=0}^{2^j-1} \sum_{m=0}^{2^j-1} n v v_0^i(n, m)}{\sum_{n=0}^{2^j-1} \sum_{m=0}^{2^j-1} v v_0^i(n, m)} - \frac{\sum_{n=0}^{2^j-1} \sum_{m=0}^{2^j-1} n v v^i(n, m)}{\sum_{n=0}^{2^j-1} \sum_{m=0}^{2^j-1} v v^i(n, m)}.$$

Для группы $G_{c,y}$ –

$$WF_C(B) = WB(x, y + \Phi w^i_2),$$

где

$$\Phi w^i_2 = \frac{\sum_{n=0}^{2^j-1} \sum_{m=0}^{2^j-1} m v v_0^i(n, m)}{\sum_{n=0}^{2^j-1} \sum_{m=0}^{2^j-1} v v_0^i(n, m)} - \frac{\sum_{n=0}^{2^j-1} \sum_{m=0}^{2^j-1} m v v^i(n, m)}{\sum_{n=0}^{2^j-1} \sum_{m=0}^{2^j-1} v v^i(n, m)}.$$

Для группы $G_{c,x,y}$ –

$$WF_C(B) = WB(x + \Phi w^i_1, y + \Phi w^i_2),$$

где

$$\Phi w^i_1 = \frac{\sum_{n=0}^{2^j-1} \sum_{m=0}^{2^j-1} n v v_0^i(n, m)}{\sum_{n=0}^{2^j-1} \sum_{m=0}^{2^j-1} v v_0^i(n, m)} - \frac{\sum_{n=0}^{2^j-1} \sum_{m=0}^{2^j-1} n v v^i(n, m)}{\sum_{n=0}^{2^j-1} \sum_{m=0}^{2^j-1} v v^i(n, m)},$$

$$\Phi w^i_2 = \frac{\sum_{n=0}^{2^j-1} \sum_{m=0}^{2^j-1} m v v_0^i(n, m)}{\sum_{n=0}^{2^j-1} \sum_{m=0}^{2^j-1} v v_0^i(n, m)} - \frac{\sum_{n=0}^{2^j-1} \sum_{m=0}^{2^j-1} m v v^i(n, m)}{\sum_{n=0}^{2^j-1} \sum_{m=0}^{2^j-1} v v^i(n, m)}.$$

Для группы G_m –

$$WF_d(B) = WB(x \Phi w^i, y \Phi w^i),$$

$$WF_d(B) = WB(x \Phi_1 w^i, y \Phi_2 w^i),$$

где

$$\Phi w^i = \frac{\sum_{n=0}^{2^j-1} \sum_{m=0}^{2^j-1} v v^i(n, m)}{\sum_{n=0}^{2^j-1} \sum_{m=0}^{2^j-1} v v_0^i(n, m)}, \quad \Phi_1 w^i = \sqrt{\frac{\sum_{n=0}^{2^j-1} \sum_{m=0}^{2^j-1} n^2 v v^i(n, m)}{\sum_{n=0}^{2^j-1} \sum_{m=0}^{2^j-1} v v^i(n, m)}},$$

$$\Phi_2 w^i = \sqrt{\frac{\sum_{n=0}^{2^j-1} \sum_{m=0}^{2^j-1} m^2 v v^i(n, m)}{\sum_{n=0}^{2^j-1} \sum_{m=0}^{2^j-1} v v^i(n, m)}}.$$

Для группы $G_{h,x}$ –

$$WF_{Hx}(B) = WB(x + \Phi w^i y, y),$$

где

$$\Phi w^i = \frac{\sum_{n=0}^{2^j-1} \sum_{m=0}^{2^j-1} nm v v^i(n, m)}{\sum_{n=0}^{2^j-1} \sum_{m=0}^{2^j-1} m^2 v v^i(n, m)}$$

Для группы $G_{h,y}$ –

$$WF_{Hy}(B) = WB(x, y + \Phi w^i x),$$

где

$$\Phi w^i = \frac{\sum_{n=0}^{2^j-1} \sum_{m=0}^{2^j-1} nm v v^i(n, m)}{\sum_{n=0}^{2^j-1} \sum_{m=0}^{2^j-1} m^2 v v^i(n, m)}.$$

Для группы G_u –

$$F_U(B) = B(x \sin \Phi w^i + y \cos \Phi w^i, -x \sin \Phi w^i + y \cos \Phi w^i),$$

где

$$\Phi w^i = \frac{1}{2} \arctg \frac{\sum_{n=0}^{2^j-1} \sum_{m=0}^{2^j-1} 2v v^i(n, m)(n^3 m + nm^3)}{\sum_{n=0}^{2^j-1} \sum_{m=0}^{2^j-1} v v^i(n, m)(n^4 + m^4)}.$$

Для группы G_p –

$$F_P(B) = B\left(\frac{x}{\Phi w^i y + 1}, \frac{y}{\Phi w^i y + 1}\right),$$

где

$$\Phi w^i = \frac{\sum_{n=0}^{2^j-1} \sum_{m=0}^{2^j-1} \frac{v v^i(n, m)}{n^4}}{\sum_{n=0}^{2^j-1} \sum_{m=0}^{2^j-1} \frac{v v^i(n, m)}{n^3}}.$$

2. Вычисление коэффициентов разложения изображения

Произведем двумерное вейвлет-преобразование контура изображения. В результате преобразования мы получим ряд коэффициентов c_i . Ввиду громоздкости вывода на экран величины коэффициентов разложения не приводятся.

Используя полученные коэффициенты по формуле обратного двумерного вейвлет-преобразования, можно восстановить контур изображения в полном объеме.

Полученные разложения вычисляются легко и быстро, время вычисления линейно и зависит только от размера изображения. Погрешность восстановления изображения составляет $\text{ans} = 1,7053e-0,13$, так что в данном случае можно считать восстановление изображения полным.

Следующим шагом для упрощения формы описания контура изображения при помощи вейвлет преобразования является частичное обнуление коэффициентов, величина которых ничтожно мала. Обнуление предельно малых значений коэффициентов позволяет также устранить шум или помехи в изображении (контуре). Вследствие обнуления коэффициентов существенного изменения в описании контура изображения не произойдет. Таким образом, контур изображения будет описан в компактной цифровой форме.

Описанный выше метод можно использовать в задачах распознавания образов и изображений.

На современном этапе для поиска соответствия изображений в базе данных используются методы поиска по ключевым словам описываемого изображения (цветовые гистограммы, текстурный анализ, анализ формы изображения, а также совокупность этих методов). Такой метод имеет свои недостатки, связанные с затруднительным описанием свойств объектов.

Рассмотрим теперь входное нормализованное изображение (контур) D . Путем двумерного вейвлет-преобразования получим матрицу коэффициентов $Dc_i[i, j]$ анализируемого изображения. Матрица $Dc_i[i, j]$ отражает ничто иное, как контур изображения. Для распознавания контура изображения необходимо теперь сравнить его с базой данных эталонов изображений (контуров). База данных эталонов изображений $Bc_i[i, j]$ (контуров) представлена в виде коэффициентов, полученных в ходе двухмерного вейвлет преобразования. В случае, когда $Dc_i[i, j]$ соответствует $Bc_i[i, j]$, можно считать, что изображение распознано.

Вычисляя разницу

$$\|D - B\| = \sum_{i, j} |Dc_i[i, j] - Bc_i[i, j]|$$

и погрешность, можно также идентифицировать исследуемое изображение.

Использование данного метода для описания изображения (контура) позволяет производить поиск изображения в базе данных также при любом разрешении исследуемого изображения.

Для упрощения вычислений и достижения максимальной скорости в анализе изображения можно рассматривать только те коэффициенты $Dc_i[i, j]$, которые не равны 0.

Выводы

В статье предложено использовать уровни вейвлет-преобразования для выделения контура изображения. Метод позволяет сократить время выделения контура изображения.

Основной трудностью теории и практического применения метода является относительная сложность, так как методы вейвлет-преобразований основываются на современных результатах функ-

ционального анализа, теории функций и вычислительной математики.

Использование метода позволяет сравнивать контур нормализованного изображения с контуром эталонного изображения независимо от разрешения поступающего изображения.

Список литературы: 1. *Путятин, Е.* Обработка изображений в робототехнике [Текст] / Е. Путятин, С. Аверин; – М.: Машиностроение, 1990. – 320 с. 2. *Петухов, А.* Введение в теорию базисов всплесков [Текст] / А. Петухов. – СПб.: Изд-во СПбГТУ, 1999. – 132 с. 3. *Малла, С.* Вэйвлеты в обработке сигналов [Текст] / С. Малла. – М.: Мир, 2005. – 671 с. 4. *Форсайт, Д.* Компьютерное зрение. Современный подход [Текст] / Д. Форсайт, Ж. Понс; – М.: Издательский дом «Вильямс», 2004. – 928 с. 5. *Прэтт, У.* Цифровая обработка изображений [Текст] / У. Прэтт. – М.: Мир, 1982. Кн.1 – 312 с. 6. *Хорн, Б.* Зрение роботов [Текст] / Б. Хорн. – М.: Мир, 1989. – 487 с.

Поступила в редколлегию 26.03.2010 г.

УДК 681.3.07

Використання рівнів вейвлет-перетворення для виділення контуру зображення / В.А. Любченко, О.А. Кобилін // Біоніка інтелекту: наук.-техн. журнал. – 2010. – № 1 (72). – С. 89–93.

Використання різних рівнів вейвлет-перетворення зображення дозволяє скоротити час необхідний для виділення контура зображення, а також скоротити час нормалізації в системах технічного зору.

Л. 3. Бібліогр.: 6 найм.

UDK 681.3.07

Using wavelet-transformation levels for image contours detection / V.A. Lyubchenko, O.A. Kobylin // Bionics of Intelligence: Sci. Mag. – 2010. – № 1 (72). – P. 89–93.

Using different levels of image wavelet-transformation allows to decrease the time necessary for image contours detection and also to reduce the time of normalization in the technical vision systems.

Fig. 3. Ref.: 6 items.

УДК 519.7



МОДЕЛЬ УСТНОЙ РЕЧИ

Н. Е. Русакова¹

¹ХНУРЭ, г. Харьков, Украина

Статья посвящена разработке методов формального описания структуры естественного языка с помощью алгебры предикатов и алгебры предикатных операций, а также аппарата логических сетей; рассматривается модель устного языка на примере звукового варианта склонения полных непряжательных имен прилагательных. Актуальность этой темы определяется перспективностью применения полученных методов для разработки систем общения с компьютером на естественном языке.

АЛГЕБРА ПРЕДИКАТОВ, ЛОГИЧЕСКАЯ СЕТЬ, ЕСТЕСТВЕННЫЙ ЯЗЫК, ПАРАДИГМАТИЧЕСКАЯ ТАБЛИЦА

Введение

В настоящее время все более совершенствуется компьютерная обработка естественного языка. Лингвистические труды содержат неформализованные описания морфологических явлений, которые непросто перевести на математический язык. Эти описания адресованы для понимания человеку. Математические же описания морфологии языка адресованы вычислительной машине, они должны быть понятны ей. В качестве объектов математического описания будем рассматривать тексты, то есть представленные в виде строчек знаковые речевые сообщения, которыми люди обмениваются в процессе интеллектуальной деятельности. Важно то, что далеко не любая строчка, составленная из знаков, может быть признана текстом [1].

Целью работы является развитие формальных средств описания естественного языка – разработка методов алгебрологического аппарата логических сетей, а также их применение для формализации морфологических структур на примере построения логической сети для звукового варианта склонения полных непряжательных имен прилагательных.

1. Звуковой вариант склонения полных непряжательных имен прилагательных

В статье [2] для примера рассматривается, формально описывается на языке алгебры предикатов и доводится до воспроизведения на логической сети механизм склонения полных непряжательных имен прилагательных. Здесь предлагается письменный вариант, который более прост по сравнению со звуковым. Это обусловлено тем, что звуковая речь более древняя, чем письменная и вполне может обходиться без письменной, в то время как письменная без звуковой обойтись не может. Звуковой вариант предельно регулярен, правила его достаточно просты, этот механизм языка уникален, благодаря своей простоте и привычности для любого человека. Исследование письменного варианта склонения имён прилагательных невозможно без привлечения факторов звуковой речи (ударность-безударность, твердость-мягкость). Исследование же звукового варианта возможно без опоры на элементы письменного языка. Предшественники до

сих пор поступали наоборот, рассматривали письменный вариант, а звуковым вообще не интересовались. Грамматика русского языка как академическая, так и школьная, строятся по следующему принципу: вначале немного говорится о звуковом варианте речи, а затем всё описание механизма языка ведется для варианта письменной речи.

Для рассмотрения выбираем полные непряжательные имена прилагательные. Достаточно основательная проработка этого раздела производится в языкознании. Пряжательные прилагательные сложнее, поскольку имеют беглость, например *лисий-лисья*, и несколько типов склонения. В полных непряжательных прилагательных всего лишь один тип склонения, имеющий семь четких вариантов склонения. Склонение полных непряжательных имен прилагательных образуют в грамматике компактный раздел морфологии языка [3]. Склонение происходит с помощью парадигматической таблицы, которая для звукового варианта представлена табл. 1.

Таблица 1

Падеж, x_3	Число, x_2				Множественное
	Единственное				
	Род, x_1				
	мужской	женский	средний		
именительный	<i>слабый_</i>	<i>слабая</i>	<i>слабое</i>	<i>слабые</i>	
родительный	<i>слабово</i>	<i>слабой_</i>	<i>слабово</i>	<i>слабых_</i>	
дательный	<i>слабому</i>	<i>слабой_</i>	<i>слабому</i>	<i>слабым</i>	
винительный	<i>слабый_</i>	<i>слабую</i>	<i>слабое</i>	<i>слабые</i>	
	<i>слабово</i>			<i>слабых_</i>	
творительный	<i>слабым_</i>	<i>слабой_</i>	<i>слабым_</i>	<i>слабыми</i>	
		<i>слабую</i>			
предложный	<i>слабом_</i>	<i>слабой</i>	<i>слабом_</i>	<i>слабых_</i>	

Вводим необходимые для формального описания предметные переменные: x_4 – признак одушевленности формы слова со значениями O – одушевленный, H – неодушевленный; x_1 – род формы слова со значениями M – мужской, J – женский, C – средний; x_2 – число формы слова со значениями E – единственное, M – множественное; x_3 – падеж формы слова со значениями I – именительный, P – родительный, D – дательный, V – винительный, T – творительный, Π – предложный.

В ячейках таблицы указаны формы слова в зависимости от их рода x_1 , числа x_2 и падежа x_3 . Таблица составлена для конкретного слова *слабый*.

Для полных имен прилагательных имеются типы склонений [3]. Ниже приведены таблицы окончаний (табл. 2-7) для всех типов склонения с указанием их номеров, которые сопровождаются примерами слов, склоняемых по данному типу.

В звуковом варианте получилось меньше типов склонения – не 7, а 6. Исчез 4-й тип склонения. Если ударная основа оканчивается на звуки *ж, ш*, то слово склоняется по 3-му типу. Если основа оканчивается на *ч, щ*, то слово склоняется по 5-му типу. В парадигматических таблицах определяем для каждого склонения переменные: y_1 – последний звук основы слова, y_2 – признак ударности-безударности, y_3 – мягкости-твердости, u – первичный класс последнего звука основы слова.

Таблица 2

1 – слабый			
ый_	айя	ойе	ыйе
ово	ой_	ово	ых_
ому	ой_	ому	ыми
ый_	уйю	ойе	ыйе
ово			ых_
ым_	ой_	ым_	ыми
	ойю		
ом_	ой_	ом_	ых_
$y_1 \in \{б, в, д, з, л, м, н, т, н, р, с\};$ $y_2 = у; y_3 = т; u = б$			

Таблица 3

2 – чужой			
ой_	айя	ойе	ыйе
ово	ой_	ово	ых_
ому	ой_	ому	ыми
ой_	уйю	ойе	ыйе
ово			ых_
ым_	ой_	ым_	ыми
	ойю		
ом_	ой_	ом_	ых_
$y_1 \in \{б, в, д, з, м, н, т, н, р, с, ж, ш\};$ $y_2 = б; y_3 = т; u = ж$			

Таблица 4

3 – горячий			
ий_	айя	ейе	ийе
ево	ей_	ево	их_
ему	ей_	ему	ими
ий_	юйю	ейе	ыйе
ево			их_
им_	ей_	им_	ими
	ейю_		
ем	ей_	ем_	их_
$y_1 \in \{н, р, с, ч, ш, _\}; y_2 = у; y_3 = м; u = ч$			

Рассмотрим формирование первичных, вторичных и третичных классов последнего звука основы слова.

Первичные классы:

$$\begin{aligned}
 y_1^б \vee y_1^в \vee y_1^д \vee y_1^з \vee y_1^л \vee y_1^м \vee y_1^н \vee y_1^т \vee y_1^р \vee y_1^с &= u^б; \\
 y_1^н \vee y_1^п \vee y_1^к &= u^н; \\
 y_1^{жс} \vee y_1^{шс} &= u^{жс}; \\
 y_1^ш &= u^ш; \\
 y_1^ч \vee y_1^{щ} \vee y_1^ц &= u^ч; \\
 y_1^г \vee y_1^к \vee y_1^х &= u^г.
 \end{aligned}$$

Вторичные классы:

$$\begin{aligned}
 u^{жс} &= v^{жс}; \\
 u^{жс} \vee u^ш &= v^ш; \\
 u^б \vee u^н &= v^б; \\
 u^н \vee u^ч &= v^ч; \\
 u^г &= v^г.
 \end{aligned}$$

Таблица 5

4 – строгий			
ий_	айя	ойе	ийе
ево	ой_	ово	их_
ому	ой_	ому	ими
ий_	уйю	ойе	ийе
его			их_
им_	ой_	им_	ими
	ойю		
ом_	ой_	ом_	их_
$y_1 \in \{г, к, х\}; y_2 = у; y_3 = м; u = г$			

Таблица 6

5 – тугой			
ой_	айя	ойе	ийе
ово	ой_	ово	их_
ому	ой_	ому	ими
ой_	уйю	ойе	ийе
ово			их_
им_	ой_	им_	ими
	ойю		
ом_	ой_	ом_	их_
$y_1 \in \{г, к, х\}; y_2 = б; y_3 = т; u = г$			

Таблица 7

7 – куцый			
ый_	айя	эйе	ыйе
эво	эй_	эво	ых_
эму	эй_	эму	ыми
ый_	уйю	эйе	ыйе
эво			ых_
ым_	эй_	ым_	ыми
	эйю_		
эм_	эй_	эм_	ых_
$y_1 \in \{ж, щ, ц\}; y_2 = у; y_3 = т; u = ц$			

Третичные классы:

$$\begin{aligned} v^b &= w^b; \\ v^b \vee v^{жс} &= w^{жс}; \\ v^u &= w^u; \\ v^e &= w^e; \\ v^y &= w^y. \end{aligned}$$

Производим нумерацию ячеек парадигматической таблицы. Полученные номера ячеек в дальнейшем будут использованы для обозначения влияний контекста. Некоторые из влияний контекста получат более одного имени. Однако от этого недостатка мы впоследствии постараемся избавиться. Для каждой ячейки парадигматической таблицы вводим свой номер r (табл.8).

Таблица 8

	x_2	ME	$ЖЕ$	$СЕ$	$-M$	
x_3	$И$	1	8	15	21	
	$Р$	2	9	16	22	
	$Д$	3	10	17	23	
	$В$	4	11	18	24	x_4
		5			25	
	$Т$	6	12	19	26	x_5
			13			
	$П$	7	14	20	27	

Выразим номера r ячеек парадигматической таблицы через признаки $x_1 - x_5$.

Преобразование блока контекста слова имеет следующий вид:

$$\begin{aligned} x_1^M x_2^E x_3^И &= r^1; x_1^M x_2^E x_3^Р &= r^2; x_1^M x_2^E x_3^Д &= r^3; \\ x_1^M x_2^E x_3^В &= r^4; x_1^M x_2^E x_3^О &= r^5; x_1^M x_2^E x_3^Т &= r^6; \\ x_1^M x_2^E x_3^П &= r^7; x_1^Ж x_2^E x_3^И &= r^8; x_1^Ж x_2^E x_3^Р &= r^9; \\ x_1^Ж x_2^E x_3^Д &= r^{10}; x_1^Ж x_2^E x_3^В &= r^{11}; x_1^Ж x_2^E x_3^Т &= r^{12}; \\ x_1^Ж x_2^E x_3^О &= r^{13}; x_1^Ж x_2^E x_3^П &= r^{14}; x_1^С x_2^E x_3^И &= r^{15}; \\ x_1^С x_2^E x_3^Р &= r^{16}; x_1^С x_2^E x_3^Д &= r^{17}; x_1^С x_2^E x_3^В &= r^{18}; \\ x_1^С x_2^E x_3^Т &= r^{19}; x_1^С x_2^E x_3^П &= r^{20}; x_2^M x_3^И &= r^{21}; \\ x_2^M x_3^Р &= r^{22}; x_2^M x_3^Д &= r^{23}; x_2^M x_3^В &= r^{24}; \\ x_2^M x_3^О &= r^{25}; x_2^M x_3^Т &= r^{26}; x_2^M x_3^П &= r^{27}. \end{aligned}$$

Составляем таблицу типов склонений в зависимости от признаков y_2, y_3 , где s – тип склонения.

Таблица 9

w	$УТ$	$УМ$	$БТ$	$БМ$
\bar{b}	1	—	—	—
$\bar{жс}$	—	—	2	—
$\bar{ч}$	—	3	—	—
$\bar{ц}$	6	—	—	—
$\bar{з}$	—	4	5	—

В итоге получаем блок слова, имеющий вид:

$$\begin{aligned} w^b y^y y^T &= s^1; w^{жс} y^b y^T &= s^2; w^y y^y y^M &= s^3; \\ w^e y^y y^M &= s^4; w^e y^b y^T &= s^5; w^y y^y y^T &= s^6, \end{aligned}$$

который зависит от последнего звука основы слова и признаков ударности-безударности и твердости-мягкости.

Связь внешних переменных представлена в виде сети (рис. 1).

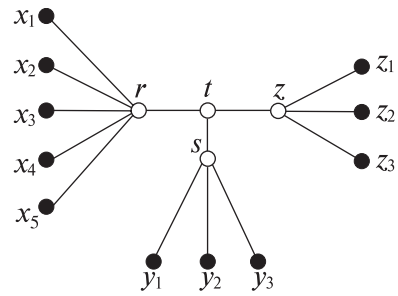


Рис. 1. Логическая сеть

2. Классификация первых звуков окончания

Механизм языка иерархичен, следовательно, исследование языка необходимо вести в определенном порядке [4]. Механизм устной речи, появляющийся при склонении имён прилагательных, можно описать, не привлекая письменную речь, в то время как для изучения письменной речи нужны признаки смягчения и ударности звука. Если изучать сначала письменный вариант склонения имён прилагательных, то создаётся впечатление, что буквы e и o противопоставлены друг другу: $om-em, ogo-ego$ и т.д. Но если рассмотреть сначала устный вариант, то становится ясным, что звук e противопоставлен звуку \bar{e} , а звук o – звуку \bar{o} . Звук \bar{e} из письменного варианта склонения прилагательных вовсе исключен. Но необходимость в звуке \bar{e} возникает при рассмотрении потенциального варианта письменной речи для склонения прилагательных. После изучения устного варианта речи легче понять её письменный вариант и глубже его изучить. Гласные звуки явно соединяются в пары родственных звуков, которые противопоставлены друг другу по признаку y_3 (твердый-мягкий). Появилась необходимость выделить пять звуков. Обозначим их переменной L со значениями $a, o, y, \bar{y}, \bar{e}$. Классификация этих звуков представлена в табл. 10.

Таблица 10

y_3	a	o	y	\bar{y}	\bar{e}	L
T	a	o	y	\bar{y}	\bar{e}	z_1
M	\bar{y}	\bar{e}	\bar{y}	\bar{y}	e	

В соответствии с классификацией первого звука окончания можно произвести классификацию окончания в целом (табл. 11).

Таблица 11

T	$\bar{a}йя$	$\bar{y}юю$	$\bar{y}ми$	$\bar{e}йе$	$\bar{o}йе$	$\bar{y}йе$	$\bar{y}й_$	$\bar{o}й_$	$\bar{e}й_$
M	$\bar{y}йя$	$\bar{y}юю$	$\bar{y}ми$	$\bar{e}йе$	$\bar{o}йе$	$\bar{y}йе$	$\bar{y}й_$	$\bar{o}й_$	$\bar{e}й_$
T	$\bar{e}во$	$\bar{o}во$	$\bar{e}му$	$\bar{o}му$	$\bar{y}м_$	$\bar{e}м_$	$\bar{o}м_$	$\bar{y}х_$	
M	$\bar{e}во$	$\bar{o}во$	$\bar{e}му$	$\bar{o}му$	$\bar{y}м_$	$\bar{e}м_$	$\bar{o}м_$	$\bar{y}х_$	
T	$\bar{e}йю$	$\bar{o}йю$							
M	$\bar{e}йю$	$\bar{o}йю$							

Окончание z расчленяется естественным образом на левую (z_l) и правую (z_r) части. Основанием для разделения окончания на левую и правую части является то, что правая часть окончания определяется исключительно влиянием контекста (признаки $x_1 - x_5$), тогда как левая часть окончания определяется, кроме того, еще и типом склонения слова s . Левая часть окончания состоит из первой буквы окончания. Правая же часть окончания состоит из двух компонентов – второй и третьей букв окончания.

Теперь рассмотрим зависимость правой части окончания от первого звука окончания (табл. 12), а также расчленим правую часть окончания на звуки (табл. 13).

Таблица 12

z_1	<i>йа</i>	<i>йю</i>	<i>ми</i>	<i>йе</i>	<i>й_</i>	<i>во</i>	<i>му</i>	<i>м_</i>	<i>х_</i>
а	<i>айя</i>	—	—	—	—	—	—	—	—
у	—	<i>уйю</i>	—	—	—	—	—	—	—
ы	—	—	<i>ыми</i>	<i>йие</i>	<i>йй_</i>	—	—	<i>ым_</i>	<i>ых_</i>
э	—	<i>эйю</i>	—	<i>эие</i>	<i>эй_</i>	<i>эво</i>	<i>эму</i>	<i>эм_</i>	—
о	—	<i>ойю</i>	—	<i>оие</i>	<i>ой_</i>	<i>ово</i>	<i>ому</i>	<i>ом_</i>	—

Таблица 13

$z_2 z_3$	я	ю	и	е	_	о	у
й	<i>йя</i>	<i>йю</i>	—	<i>йе</i>	<i>й_</i>	—	—
м	—	—	<i>ми</i>	—	<i>м_</i>	—	<i>му</i>
в	—	—	—	—	—	<i>во</i>	—
х	—	—	—	—	<i>х_</i>	—	—

Проведя классификацию звуков окончания и рассмотрев связь звуков между собой, можно записать в формульном виде эту связь.

$$\begin{aligned}
 z^1 &= z_1^a z_2^{\dot{u}} z_3^{\dot{a}}; z^2 = z_1^{\dot{a}} z_2^{\dot{u}} z_3^{\dot{a}}; \\
 z^3 &= z_1^y z_2^{\dot{u}} z_3^{\dot{y}}; z^4 = z_1^{\dot{y}} z_2^{\dot{u}} z_3^{\dot{y}}; \\
 z^5 &= z_1^{b\dot{u}} z_2^m z_3^u; z^6 = z_1^u z_2^m z_3^u; \\
 z^7 &= z_1^{\dot{e}} z_2^{\dot{u}} z_3^e; z^8 = z_1^e z_2^{\dot{u}} z_3^e; \\
 z^9 &= z_1^o z_2^{\dot{u}} z_3^e; z^{10} = z_1^{b\dot{u}} z_2^{\dot{u}} z_3^e; \\
 z^{11} &= z_1^u z_2^{\dot{u}} z_3^e; z^{12} = z_1^{b\dot{u}} z_2^{\dot{u}} z_3^e; \\
 z^{13} &= z_1^u z_2^{\dot{u}} z_3^e; z^{14} = z_1^{\dot{e}} z_2^{\dot{u}} z_3^e; \\
 z^{15} &= z_1^e z_2^{\dot{u}} z_3^e; z^{16} = z_1^o z_2^{\dot{u}} z_3^e; \\
 z^{17} &= z_1^{\dot{e}} z_2^{\dot{u}} z_3^e; z^{18} = z_1^e z_2^{\dot{u}} z_3^e; \\
 z^{19} &= z_1^o z_2^{\dot{u}} z_3^e; z^{20} = z_1^{\dot{e}} z_2^m z_3^y; \\
 z^{21} &= z_1^e z_2^m z_3^y; z^{22} = z_1^o z_2^m z_3^y; \\
 z^{23} &= z_1^{b\dot{u}} z_2^m z_3^y; z^{24} = z_1^u z_2^m z_3^y; \\
 z^{25} &= z_1^{\dot{e}} z_2^m z_3^y; z^{26} = z_1^e z_2^m z_3^y; \\
 z^{27} &= z_1^o z_2^m z_3^y; z^{28} = z_1^{b\dot{u}} z_2^x z_3^y; \\
 z^{29} &= z_1^u z_2^x z_3^y; z^{30} = z_1^{\dot{e}} z_2^{\dot{u}} z_3^{\dot{y}}; \\
 z^{31} &= z_1^e z_2^{\dot{u}} z_3^{\dot{y}}; z^{32} = z_1^o z_2^{\dot{u}} z_3^{\dot{y}}.
 \end{aligned}$$

Выводы

Для построения моделей механизмов естественного языка нужны более мощные алгебрологический аппарат и методы формального описания, чем те, которые разработаны к настоящему времени. Поэтому в статье уделяется внимание описанию применяемых в этой модели существующих алгебрологических средств и разработке новых. В работе алгебрологические методы не только применяются на практике, но и развиваются далее до того уровня, при котором моделирование механизмов языка становится возможным. Главными ограничивающими факторами при моделировании языка являются недостаточное развитие алгебрологических средств и недостаточное развитие методов физического (объективного) исследования механизмов естественного языка.

В процессе рассмотрения методов можно делать вывод, что мы разрабатываем не актуальный, а потенциальный вариант устного склонения прилагательных. Потенциальный вариант разделяется на ближний (псевдослова), более далекий (введение звука ϕ в конце слова) и еще более далёкий (введение ϵ в качестве первого звука окончания).

В дальнейшем предполагается исследование совершенной дизъюнктивной нормальной формы предиката модели звукового варианта склонения полных непряжательных имен прилагательных.

Список литературы: 1. Широков, В. А. Феноменология лексикографических систем [Текст] / В. А. Широков. – К.: Наукова думка, 2004. – 327 с. 2. Бондаренко, М. Ф. Модели языка [Текст] / М. Ф. Бондаренко, Ю. П. Шабанов-Кушнарченко // Бионика интеллекта: научн.-техн. журнал. – 2004. – № 1. – С. 27-37. 3. Бархударов, С. Г. Учебник русского языка [Текст] / С. Г. Бархударов, С. Е. Крючков. – М.: Просвещение, 1970. – 296 с. 4. Зализняк, А. А. Грамматический словарь русского языка [Текст] / А. А. Зализняк. – М.: Русский язык, 1977. – 879 с. 5. Бондаренко, М. Ф. Обионике интеллекта [Текст] / М. Ф. Бондаренко, Ю. П. Шабанов-Кушнарченко // Бионика интеллекта научн.-техн. журнал. – 2006 – № 1. – С. 3-5.

Поступила в редколлегию 29.03.2010 г.

УДК 519.7

Модель усной мови / Н. Є. Русакова // Біоніка інтелекту: наук.-техн. журнал. – 2010. – № 1 (72). – С. 94–97.

В статті розглядається проблема формалізації поняття тексту та його математичного описання, тобто представлення тексту у вигляді рядків знакових мовних повідомлень, якими люди обмінюються в процесі інтелектуальної діяльності. Розглядається модель усної мови на прикладі звукового варіанта відмінювання прикметників.

Табл. 13. Іл. 1. Бібліогр.: 5 найм.

UDC 519.7

Model of verbal speech / N. E. Rusakova // Bionics of Intelligence: Sci. Mag. – 2010. – № 1 (72). – P. 94–97.

In the article the problem of formalization of concept of text is examined and him mathematical description, that presentation of text is as lines of sign linguistic reports, which people change in the press of intellectual activity. The model of verbal language is examined on the example of voice variant of conjugation of adjectives.

Tab. 13. Fig. 1. Ref.: 5 items.

УДК 681.142.2; 622.02.658.284; 621.325



ЗАДАЧА ФАКТОРИЗАЦІЇ ПРОСТОРІВ, ВИЗНАЧЕНИХ НА ТОПОЛОГІЯХ ЗОБРАЖЕНЬ

Д.Д. Пелешко¹, Ю.М.Рашкевич²

¹ НУ ЛП, м. Львів, Україна, peleshko@polynet.lviv.ua;

² НУ ЛП, м. Львів, Україна, rashkev@polynet.lviv.ua

Формалізовано задачу факторизації топологічного колірному простору рисунка та набору рисунків. Сформульовано поняття майже фактор простору та сформульовано задачу майже факторизації топологічного колірному простору зображень та наборів зображень.

ЗОБРАЖЕННЯ, НАБІР ЗОБРАЖЕНЬ, ТОПОЛОГІЯ, ФАКТОР ПРОСТІР, ФАКТОРИЗАЦІЯ, МЕТРИКА, НАПІВМЕТРИКА, МНОЖИНА МІРИ НУЛЬ, МАЙЖЕ ФАКТОРИЗАЦІЯ

Вступ

Дуже часто на практиці вирішення задачі обробки зображення розпочинається із визначення дій при заданому розбитті зображення. При цьому питання існування цього розбиття, його повнота і інше практично не розглядаються. Проте побудова та існування розбиття є окремою топологічною задачею формування покриття, а в загальному випадку – організації топологій. Іншими словами це є топологічна задача, яка потребує власного обґрунтування. І її успішне вирішення багато в чому визначає успішність вирішення подальших прикладних задач.

Для методів практичної обробки сигналів окрім задачі формування топології є ще одна важлива задача, а саме – операції над цим покриттям. Тут найважливішою задачею виступає задача факторизації за заданим відношенням еквівалентності простору покриття зображення.

1. Постановка задачі

Метою даної роботи є формалізація задачі факторизації колірному топологічного простору зображення та наборів зображень для подальшої розробки прикладних методів цифрової обробки зображень та наборів зображень.

Для досягнення цієї мети потрібно ввести метрику і дослідити її властивості, зокрема здатність виражати відношення еквівалентності.

2. Представлення рисунка та набору рисунків

Дискретне представлення кожного цифрового зображення P (надалі просто зображення) є відображенням скінченного дискретного набору значень з $\mathbb{N}^{2,+}$. Функцію C можна записати у вигляді

$$C : \mathbb{X}^{2,+d} \rightarrow Q^d, \quad (1)$$

де $\mathbb{X}^{2,+d} \subset \mathbb{N}^{2,+}$ – двовимірний топологічний многовид [6] як хаусдорфовий топологічний підпростір [1, 10] евклідового простору $\mathbb{N}^{2,+}$ [9, 10]; $Q^d \subset Color$ – замкнені обмежені підмножини просторів $\mathbb{N}^{2,+}$ та $Color$ відповідно, які є вимірними за Жорданом [1, 9]. Зважаючи на зліченність просторів \mathbb{N} , $\mathbb{N}^{2,+}$, $\mathbb{X}^{2,+d}$ та Q^d є також зліченими [1, 9].

Область визначення функції C , а саме підмножину $\mathbb{X}^{2,+d}$, будемо називати координатним (піксельним) представленням рисунка, а область значень Q^d – колірним. Фактично рисунок P є набором значень кольору (результату дії функції C) в точках-пікселях області $\mathbb{X}^{2,+d}$ із власними невід’ємними координатами.

Фізичні розміри рисунка P визначаються розмірами області $\mathbb{X}^{2,+d}$. У більшості випадків область $\mathbb{X}^{2,+d}$ є прямокутником. Тому її можна подати у вигляді

$$\mathbb{X}^{2,+d} = \{dtp_{i,j} = (i,j) \mid i = \overline{1,l} \wedge j = \overline{1,h}\}, \quad (2)$$

де $l, h \in \mathbb{N}^+$ – довжина та висота рисунка P .

У разі розгляду набору з N рисунків (1) та (2) видозмінюються як:

$$\mathbf{P} = \left\{ P_z : P_z = C_z(\mathbb{X}_z^{2,+d}) \right\}_{z=1..N}, \quad (3)$$

де

$$C_z : \mathbb{X}_z^{2,+d} \rightarrow Q_z^d, \quad z = \overline{1,N}; \quad (4)$$

$$\mathbb{X}_z^{2,+d} = \{dtp_{z,i,j} = (i,j) \mid i = \overline{1,l_z} \wedge j = \overline{1,h_z}\}, \quad (5)$$

$$\forall z : l_z, h_z \in \mathbb{N}^+.$$

Тут через \mathbf{P} позначається набір з N рисунків P_z , а $\mathbb{X}_z^{2,+d}$ – область визначення функції C_z , z -го рисунка P_z .

3. Топології зображення та наборів зображень

3.1. Формування топології в координатній області зображень та наборів зображень

Зважаючи на ”прямокутність” та дискретність $\mathbb{X}^{2,+d}$, розглянемо сім’ю підмножини $\mathbb{X}_{frm}^{2,+d} \subseteq \mathbb{X}^{2,+d}$,

$$\mathfrak{S}_{\mathbb{X}^{2,+d}} = \left\{ \mathbb{X}_{frm}^{2,+d} \mid \mathbb{X}_{frm}^{2,+d} \subseteq \mathbb{X}^{2,+d} \right\}_{m \geq 1}, \quad (6)$$

які також будуть прямокутними дискретними областями і називатимемо їх *фреймами*.

Кожен $\mathbb{X}_{frm}^{2,+d}$ визначається координатами початку (верхнього лівого кута), тобто геометричними координатами зміщення початку фрейму відносно початку зображення:

$$\begin{cases} x_{\text{поч } m} = x(\mathbf{X}_{\text{frm}}^{2+,d}) = x_{\text{поч}} + \Delta_{x m}; \\ y_{\text{поч } m} = y(\mathbf{X}_{\text{frm}}^{2+,d}) = y_{\text{поч}} + \Delta_{y m}, \end{cases} \quad (7)$$

де $x_{\text{поч } m}, y_{\text{поч } m}$ – координати початку $X^{2+,d}$; $\Delta_{x m}$ і $\Delta_{y m}$ – зміщення початку фрейму $\mathbf{X}_{\text{frm}}^{2+,d}$ відносно початку $X^{2+,d}$ в напрямках x та y відповідно. Іншими характеристиками фрейму виступають його довжина $l_{\text{frm}} = l_{X_{\text{frm}}^{2+,d}} \in \mathbb{N}^+$ та висота $h_{\text{frm}} = h_{X_{\text{frm}}^{2+,d}} \in \mathbb{N}^+$. Це означає, що $\mathbf{X}_{\text{frm}}^{2+,d}$ є деякою функцією від початку, довжини та висоти, яку зважаючи на (5) можна подати у вигляді

$$\begin{aligned} \mathbf{X}_{\text{frm}}^{2+,d} &= \mathbf{X}_{\text{frm}}^{2+,d}(x_{\text{поч } m}, y_{\text{поч } m}, l_{\text{frm}}, h_{\text{frm}}) = \\ &= \mathbf{X}_{\text{frm}}^{2+,d}(\Delta_{x m}, \Delta_{y m}, l_{\text{frm}}, h_{\text{frm}}). \end{aligned} \quad (8)$$

Набір $\mathfrak{S}_{X^{2+,d}}$ виступає топологією простору $X^{2+,d}$, а $(X^{2+,d}, \mathfrak{S}_{X^{2+,d}})$ – топологічним дискретним простором. Оскільки $\mathbf{X}_{\text{frm}}^{2+,d}$ є фреймом, то $\mathfrak{S}_{X^{2+,d}}$ будемо називати *фреймовою топологією*.

На $X^{2+,d}$ можна ввести дискретну метрику

$$\begin{aligned} d_{X^{2+,d}, \text{dis}}(dtp_{i_1, j_1}, dtp_{i_2, j_2}) &= \\ &= \begin{cases} 1, & dtp_{i_1, j_1} \neq dtp_{i_2, j_2}; \\ 0, & dtp_{i_1, j_1} = dtp_{i_2, j_2}, \end{cases} \quad dtp_{i_1, j_1}, dtp_{i_2, j_2} \in X^{2+,d}. \end{aligned} \quad (9)$$

Тоді $(X^{2+,d}, \mathfrak{S}_{X^{2+,d}})$ є метричним дискретним простором. Зазначимо, що $\mathfrak{S}_{X^{2+,d}}$ є дискретною топологією, оскільки вона індукується дискретною метрикою (9). Дискретність метрики (9) визначена в [7].

Підмножина $\mathbf{X}_{\text{frm}}^{2+,d}$ виступає *відкритою підмножиною* – *точкою простору* $(X^{2+,d}, \mathfrak{S}_{X^{2+,d}})$. Тобто топологію $\mathfrak{S}_{X^{2+,d}}$ на множині $X^{2+,d}$ задаємо системою замкнених множин.

При розгляді набору зображень можливі два випадки.

1) В загальному випадку кожен рисунок набору володіє власним топологічним простором $(X^{2+,d}, \mathfrak{S}_{X^{2+,d}})$. В такому випадку фреймову топологію набору \mathfrak{S}_P складає набір топологій

$$\mathfrak{S}_P = \{\mathfrak{S}_z\}_{z=1..N}, \quad \mathfrak{S}_z = \mathfrak{S}_{X_z^{2+,d}}, \quad (10)$$

який будемо називати *вектором фреймових топологій*. Покриття, які входять до кожного \mathfrak{S}_z , в загальному випадку є різними, але можуть і співпадати.

Якщо з кожного \mathfrak{S}_z вибрати строго по одному покриттю і сформуванати набір, то можна говорити про *фреймове покриття набору*

$$\begin{aligned} \chi_P &= \{\chi_z\}_{z=1..N}, \\ (\chi_z | N_{\chi_z}) &= \left(\chi_{X_z^{2+,d}} | N_{\chi_{X_z^{2+,d}}} \right) \subseteq \mathfrak{S}_z \in \mathfrak{S}_P, \end{aligned} \quad (11)$$

який, подібно до (10) будемо називати вектором фреймових покриттів. Зазначимо, що розмірність кожного може χ_z бути різною.

2) У другому випадку усі рисунки набору володіють однаковим координатним многовидом $X^{2+,d}$. Тоді $\mathfrak{S}_{X^{2+,d}}$ виступає фреймовою топологією набору \mathfrak{S}_P . Фактично (10) вироджується у

$$\mathfrak{S}_P = \mathfrak{S}_{X^{2+,d}}. \quad (12)$$

У другому випадку в практичних задачах (зокрема у випадках однотипних зображень) на топології $\mathfrak{S}_{X^{2+,d}}$ для кожного рисунка набору вводять власні покриття, які можуть не співпадати між собою. Іншими словами, має місце

$$\exists z_1, z_2 \in [1; N]: \chi_{z_1} \neq \chi_{z_2}, \chi_{z_1} \subseteq \mathfrak{S}_{X^{2+,d}}, \chi_{z_2} \subseteq \mathfrak{S}_{X^{2+,d}}, \quad (13)$$

де χ_{z_1} – z_1 -е покриття простору $(X^{2+,d}, \mathfrak{S}_{X^{2+,d}})$ для z_1 -го рисунка розмірністю $N_{\chi_{z_1}}$ фреймів. χ_{z_2} – аналогічно до χ_{z_1} покриття z_2 -го рисунка. У загальному випадку кількість фреймів в покриттях χ_z може не співпадати, тобто має місце $N_{\chi_{z_1}} \neq N_{\chi_{z_2}}$. Розмірність фреймів незалежно від того, чи вони належать одному покриттю чи різним покриттям фреймового покриття, χ_P також може відрізнятися

$$\exists m_1 \in [1; N_{\chi_{z_1}}] \cap [1; N_{\chi_{z_2}}]: \quad (14)$$

$$\mathbf{X}_{\text{frm } z, m}^{2+,d} \in \chi_{z_1} \wedge \mathbf{X}_{\text{frm } z, m}^{2+,d} \notin \chi_{z_2}, \quad \chi_{z_2}, \chi_{z_1} \in \chi_P.$$

Фреймове покриття набору у цьому випадку буде складатись з N різних покриттів топології $\mathfrak{S}_{X^{2+,d}}$, тобто замість (11) маємо

$$\chi_P = \left\{ \chi_z | N_{\chi_z} \right\}_{z=1..N}, \quad \chi_z \subseteq \mathfrak{S}_P = \mathfrak{S}_{X^{2+,d}}. \quad (15)$$

3.2. Формування топології наборів зображень в просторі кольору

Зважаючи на існування $(X^{2+,d}, \mathfrak{S}_{X^{2+,d}})$, (1) можна записати у вигляді

$$C: \chi' \rightarrow Q^d. \quad (16)$$

Беручи до уваги (2.7) та (3.9), оператор (3.25) можна подати у вигляді

$$\begin{aligned} C &= \left\{ C_m : \mathbf{X}_{\text{frm}}^{2+,d} \rightarrow Q_m^d \right\}_{m=1..N_{\chi}}; \\ C_m &\leftrightarrow C(\mathbf{X}_{\text{frm}}^{2+,d}); Q_m^d \subseteq Q^d, \end{aligned} \quad (17)$$

де C_m – оператор C , але визначений виключно на фреймі $\mathbf{X}_{\text{frm}}^{2+,d}$. Очевидно, що подібно до C , оператор C_m є колірним оператором. (17) визначає (16) як деяку адитивну групу операторів над $X^{2+,d}$.

Кожен C_m є неперервним оператором. Тоді рисунок P можна подати у вигляді набору фрагментів

$$\begin{aligned}
 P &\leftrightarrow \left\{ P_m \mid P_m = C_m \left(\mathbf{X}_{\text{frm}}^{2,+d} \right) \right\}_{m=1..N_\chi} = \\
 &= \left\{ \left\{ c_{m,(i,j)}^d \mid c_{m,(i,j)}^d \in Q_{m,i,j}^d \right\}_{i,j} \right\}_{m=1..N_\chi} = \\
 &= \left\{ \left\{ c_{m,(i,j)}^d \mid \forall i \in [x_{\text{поч } m} \dots x_{\text{поч } m} + l_{\text{frm}}], \right. \right. \\
 &\quad \left. \left. \forall j \in [y_{\text{поч } m} \dots y_{\text{поч } m} + h_{\text{frm}}], \right. \right. \\
 &\quad \left. \left. \exists m \in [1..N_\chi] : c_{m,(i,j)}^d \in P_m \right. \right\}_{m=1..N_\chi},
 \end{aligned} \tag{18}$$

$$P = \bigcup_{m=1}^{N_\chi} P_m, \tag{19}$$

де P_m називатимемо фрагментом зображення P .

Набір фрагментів $\{P_m\}_{m=1..N_\chi}$ утворює топологію \mathcal{U}_P (фрагментну топологію) рисунка P в колірній області, яка індукована топологією $\mathfrak{S}_{\mathbf{X}^{2,+d}}$. Це дає змогу ввести до розгляду колірний топологічний простір (P, \mathcal{U}_P) [1, 6, 15] рисунка P . При цьому (16) виступає абстрактним неперервним оператором відображення топологій

$$C : \mathfrak{S}_{\mathbf{X}^{2,+d}} \rightarrow \mathcal{U}_P. \tag{20}$$

$$P = C(\mathfrak{S}_{\mathbf{X}^{2,+d}}). \tag{21}$$

Якщо $(\mathbf{X}^{2,+d}, \mathfrak{S}_{\mathbf{X}^{2,+d}})$ визначене будь-яке покриття $(\chi_{\mathbf{X}^{2,+d}} \mid N_\chi)$, то за (18) чи (20) отримуємо покриття (фрагментне покриття)

$$\begin{aligned}
 \vartheta_P &= \{P_m \mid P_m = C(\mathbf{X}_{\text{frm}}^{2,+d}) \wedge \mathbf{X}_{\text{frm}}^{2,+d} \in \chi_{\mathbf{X}^{2,+d}}\}_{m=1..N_\chi}, \tag{22} \\
 \vartheta_P &\in \mathcal{U}_P,
 \end{aligned}$$

яке індуковане $\chi_{\mathbf{X}^{2,+d}}$. При цьому розмірність N_χ покриття $\chi_{\mathbf{X}^{2,+d}}$ є розмірністю покриття ϑ_P , і маємо $(\vartheta_P \mid N_\chi)$.

У разі існування набору (3) \mathbf{P} , подібно до фреймової топології набору \mathfrak{F}_P можливі два випадки:

1. Для усіх рисунків набору маємо (10), тобто топології кожного з рисунків набору (P, \mathcal{U}_P) індукуються різними топологіями \mathfrak{S}_z координатних просторів $\mathbf{X}_z^{2,+d}$ кожного рисунка. Тоді набір можна представити у вигляді

$$\begin{aligned}
 \mathbf{P} &= \left\{ P_z \mid P_z = C(\mathfrak{S}_z) \right\}_{z=1..N} = \\
 &= \left\{ \left\{ P_{z,m} \mid P_{z,m} = C_{z,m} \left(\mathbf{X}_{\text{frm } z,m}^{2,+d} \right) \right\}_{m=1..N_{\chi_z}} \right\}_{z=1..N} = \\
 &= \left\{ \left\{ \left\{ c_{z,m,(i,j)}^d \mid \right. \right. \right. \\
 &\quad \left. \left. \forall i \in [x_{\text{поч } z,m} \dots x_{\text{поч } z,m} + l_{\text{frm } z,m}], \right. \right. \\
 &\quad \left. \left. \forall j \in [y_{\text{поч } z,m} \dots y_{\text{поч } z,m} + h_{\text{frm } z,m}], \right. \right. \\
 &\quad \left. \left. \exists m \in [1..N_{\chi_z}] : \right. \right. \\
 &\quad \left. \left. c_{z,m,(i,j)}^d \in P_{z,m} \subset \mathcal{U}_{P_z} \right. \right\}_{m=1..N_{\chi_z}} \right\}_{z=1..N},
 \end{aligned} \tag{23}$$

де $P_{z,m}$ – m -ий фрагмент z -го рисунка P_z ; $x_{\text{поч } z,m}, y_{\text{поч } z,m}, l_{\text{frm } z,m}, h_{\text{frm } z,m}$ – координати початку m -го фрейма z -го рисунка; $\mathcal{U}_{P_z} = \mathcal{U}_z$ – топологія z -го рисунка набору \mathbf{P} ; N_{χ_z} – розмірність покриття χ_z і $\vartheta_z = \vartheta_{P_z}$.

А вектор фрагментних топологій набору \mathcal{U}_P (фрагментна топологія набору) матиме вигляд

$$\mathcal{U}_P = \left\{ \mathcal{U}_z \right\}_{z=1..N}, \quad \mathcal{U}_z = \mathcal{U}_{\mathbf{X}_z^{2,+d}}. \tag{24}$$

Оскільки кожна топологія \mathcal{U}_z індукується відповідною \mathfrak{S}_z , то можна стверджувати, що \mathcal{U}_P індукується \mathfrak{F}_P .

2. У випадку, якщо топології усіх рисунків (P_z, \mathcal{U}_{P_z}) індукуються однією топологією $\mathfrak{S}_{\mathbf{X}^{2,+d}}$, то (23) видозміниться як:

$$\begin{aligned}
 \mathbf{P} &= \left\{ P_z \mid P_z = C(\mathfrak{S}_z) \right\}_{z=1..N} = \\
 &= \left\{ \left\{ P_{z,m} \mid P_{z,m} = C_{z,m} \left(\mathbf{X}_{\text{frm } z,m}^{2,+d} \right) \right\}_{m=1..N_{\chi_z}} \right\}_{z=1..N} = \\
 &= \left\{ \left\{ \left\{ c_{z,m,(i,j)}^d \mid \right. \right. \right. \\
 &\quad \left. \left. \forall i \in [x_{\text{поч } z,m} \dots x_{\text{поч } z,m} + l_{\text{frm } z,m}], \right. \right. \\
 &\quad \left. \left. \forall j \in [y_{\text{поч } z,m} \dots y_{\text{поч } z,m} + h_{\text{frm } z,m}], \right. \right. \\
 &\quad \left. \left. \exists m \in [1..N_{\chi_z}] : \right. \right. \\
 &\quad \left. \left. c_{z,m,(i,j)}^d \in P_{z,m} \subset \mathcal{U}_{P_z} \right. \right\}_{m=1..N_{\chi_z}} \right\}_{z=1..N},
 \end{aligned} \tag{25}$$

а вектор фрагментних топологій набору \mathcal{U}_P матиме вигляд

$$\mathcal{U}_P = \left\{ \mathcal{U}_z \right\}_{z=1..N}, \quad C_z : \mathfrak{S}_{\mathbf{X}^{2,+d}} \rightarrow \mathcal{U}_z, \tag{26}$$

За обома випадками набір \mathbf{P} представляється у вигляді

$$\mathbf{P} = C(\mathfrak{F}_{\mathbf{X}^{2,+d}}), \tag{27}$$

де

$$C : \mathfrak{F}_{\mathbf{X}^{2,+d}} \rightarrow \mathcal{U}_P = \left\{ C_z : \mathfrak{S}_z \rightarrow \mathcal{U}_z \right\}_{z=1..N}. \tag{28}$$

Якщо з кожної топології вибрати \mathcal{U}_z по одному ϑ_z і сформувавши набір, то отримаємо вектор фрагментного покриття набору

$$\mathfrak{F}_P = (\mathfrak{F}_P \mid N) = \left\{ \vartheta_z \right\}_{z=1..N}, \quad \vartheta_z \in \mathcal{U}_z. \tag{29}$$

4. Задача факторизації простору покриття зображення та наборів зображень

4.1. Факторизація топології колірного простору зображення та наборів зображень

Нехай на рисунку P визначено координатну $\mathfrak{S}_{\mathbf{X}^{2,+d}}$ і колірну \mathcal{U}_P топології, в яких вибрано покриття $(\chi_{\mathbf{X}^{2,+d}} \mid N_\chi) \subseteq \mathfrak{S}_{\mathbf{X}^{2,+d}}$, $(\vartheta_P \mid N_\chi) \subseteq \mathcal{U}_P$. Приймається, що усі фрагменти P_m покриття ϑ_P мають

однакову розмірність. Для $(\vartheta_P | N_\chi)$ визначимо метрику

$$\forall P_{m_1}, P_{m_2} \in \vartheta_P : d_{r,fr}(P_{m_1}, P_{m_2}) = \left| r(P_{m_1}, P_{m_2}) - 1 \right|, \quad (30)$$

де $r(P_{m_1}, P_{m_2})$ – кореляція [] між фрагментами; P_{m_1}, P_{m_2} – фрагменти (колірні прямокутні області) зображення P .

Твердження 1.

Співвідношення (30) є метрикою простору (P, \mathcal{U}_P) .

Доведення.

◁

Нехай в топології \mathcal{U}_P задано покриття $(\vartheta_P | N_\chi) \subseteq \mathcal{U}_P$. Розглянемо умови метрики [].

1. $d_{r,fr}(P_{m_1}, P_{m_2}) \geq 0$ – впливає з означення функції модуля.

2. $d_{r,fr}(P_m, P_m) = 0$ – впливає з властивості кореляції Оскільки $r(P_m, P_m) = 1$ [], то за (30) маємо

$$d_{r,fr}(P_m, P_m) = |1 - 1| = 0. \quad (31)$$

3. Нерівність трикутника

$$d_{r,fr}(P_{m_1}, P_{m_3}) \leq d_{r,fr}(P_{m_1}, P_{m_2}) + d_{r,fr}(P_{m_2}, P_{m_3}). \quad (32)$$

Для доведення нерівності трикутника розглянемо обмеженість метрики (30). Згідно з властивістю кореляції маємо

$$0 \leq r(P_{m_1}, P_{m_3}) \leq 1. \quad (33)$$

При $|r(P_{m_1}, P_{m_3})| = 1$ маємо

$$d_{r,fr}(P_{m_1}, P_{m_3}) = |1 - 1| = 0, \quad (34)$$

а при $|r(P_{m_1}, P_{m_3})| = 0$

$$d_{r,fr}(P_{m_1}, P_{m_3}) = |0 - 1| = 1. \quad (35)$$

Тоді оцінку метрики (30) можна записати як

$$d_{r,fr}(P_{m_1}, P_{m_3}) \leq 1. \quad (36)$$

Звідси

$$d_{r,fr}(P_{m_1}, P_{m_2}) + d_{r,fr}(P_{m_2}, P_{m_3}) \leq 1 + 1 = 2. \quad (37)$$

З (36) та (37) випливає нерівність трикутника.

Отже (30) володіє усіма властивостями метрики.

▷

Маючи метрику (30) можна сформулювати задачу факторизації простору $(\vartheta_P | N_\chi)$.

Задача 1.

Факторизація покриття $(\vartheta_P | N_\chi)$ топології \mathcal{U}_P полягає у побудові фактор-простору $\vartheta_{(\vartheta_P | N_\chi), d_{r,fr}} / \sim$ за допомогою метрики (30).

Твердження 2.

(30) є напівметрикою простору (P, \mathcal{U}_P) .

Доведення.

◁

Нехай в топології \mathcal{U}_P задано покриття $(\vartheta_P | N_\chi) \subseteq \mathcal{U}_P$. Оскільки $(\vartheta_P | N_\chi)$ може містити фрагменти (наприклад P_{m_2}), кореляція яких із заданим (наприклад P_{m_1}) рівна ± 1

$$r(P_{m_1}, P_{m_2}) = \pm 1. \quad (38)$$

Тоді метрика (30) є фактично відношенням еквівалентності двох фрагментів

$$d_{r,fr}(P_{m_1}, P_{m_2}) = 0 \wedge P_{m_1} \neq P_{m_2}, \quad (39)$$

а тому є напівметрикою.

▷

Твердження 2 є підтвердженням існуванням задачі 1.

Твердження 3.

Фактор-простір $\vartheta_{(\vartheta_P | N_\chi), d_{r,fr}} / \sim$ належить топології \mathcal{U}_P .

Доведення впливає з означення фактор-простору

$$\vartheta_{(\vartheta_P | N_\chi), d_{r,fr}} / \sim \subseteq \vartheta_P \subseteq \mathcal{U}_P. \quad (40)$$

З твердження 3 випливає ще одне важливе твердження.

Твердження 4.

Факторизація простору є операцією звуження простору $(\vartheta_P | N_\chi)$.

Доведення впливає з (40). При цьому важливою є нерівність

$$N_{\chi/\sim} \leq N_\chi, \quad (41)$$

де $N_{\chi/\sim}$ – розмірність $\vartheta_{(\vartheta_P | N_\chi), d_{r,fr}} / \sim$.

Відношення (41) визначає ще одну важливу нерівність

$$\mu\left(\vartheta_{(\vartheta_P | N_\chi), d_{r,fr}} / \sim\right) \leq \mu(\vartheta_P), \quad (42)$$

де μ – зліченно-адитивна міра множини.

Розглянемо набір рисунків \mathbf{P} (27). Нехай на \mathbf{P} задано топологію \mathcal{U}_P (26). Виберемо в цій топології фрагментне покриття $(\mathfrak{P} | N)$ (29). Вважаємо, що усі фрагменти покриття $(\mathfrak{P} | N)$ мають однакову розмірність. Між фрагментами різних ϑ_z покриття $(\mathfrak{P} | N)$ за аналогією з (30) введемо метрику

$$\forall P_{z_1, m_1} \in \vartheta_{z_1}, P_{z_2, m_2} \in \vartheta_{z_2} : \quad (43)$$

$$d_{r,fr}(P_{z_1, m_1}, P_{z_2, m_2}) = \left| r(P_{z_1, m_1}, P_{z_2, m_2}) - 1 \right|.$$

Тут можлива ситуація, коли $z_1 = z_2$.

Зважаючи на (43) задачу 1 у випадку набору можна сформулювати так:

Задача 2.

Факторизація покриття $(\mathfrak{P} | N)$ топології \mathcal{U}_P полягає у побудові фактор простору $\mathfrak{P}_{(\mathfrak{P} | N), d_{r,fr}} / \sim$ за допомогою метрики (30).

4.2. Майже факторизація простору покриття рисунка та наборів рисунків

Якщо відношення еквівалентності (39) на підмножині $\vartheta_0(\vartheta_P|N_\chi)_{d_r,fr} / \sim^\varepsilon \subseteq \vartheta_P$ міри 0 прийняти таким

$$P_{m_1}, P_{m_2} \in \vartheta_0, \quad 0 < d_{r,fr}(P_{m_1}, P_{m_2}) \leq \varepsilon, \quad (44)$$

то можна ввести до розгляду майже факторизацію, або ε -факторизацію.

Простір, отриманий в результаті ε -факторизації, будемо називати майже фактор-простором, або ε -факторпростором і позначатимемо $\vartheta_{(\vartheta_P|N_\chi)_{d_r,fr}} / \sim^\varepsilon$.

Означення 1.

Майже фактор простором або ε -факторпростором будемо називати такий підпростір $\vartheta_{(\vartheta_P|N_\chi)_{d_r,fr}} / \sim^\varepsilon$ простору $(\vartheta_P | N_\chi)$, який є фактор простором усяди за виключення множини міри нуль, на якій виконується умова (44), тобто має місце

$$\varepsilon > 0 \quad \exists \vartheta_0(\vartheta_P|N_\chi)_{d_r,fr} / \sim^\varepsilon \subseteq \vartheta_{(\vartheta_P|N_\chi)_{d_r,fr}} / \sim^\varepsilon \subseteq \vartheta_P, \\ \mu(\vartheta_0) = 0: \quad 0 < d_{r,fr}(P_{m_1}, P_{m_2}) \leq \varepsilon; \quad (45)$$

$$\vartheta_{(\vartheta_P|N_\chi)_{d_r,fr}} / \sim^\varepsilon = \vartheta_{(\vartheta_P|N_\chi)_{d_r,fr}} / \sim \cup \vartheta_0(\vartheta_P|N_\chi)_{d_r,fr} / \sim^\varepsilon.$$

$$\text{Тут } P_{m_1}, P_{m_2} \in \vartheta_0(\vartheta_P|N_\chi)_{d_r,fr} / \sim^\varepsilon.$$

Твердження 5.

Міра простору $\vartheta_{(\vartheta_P|N_\chi)_{d_r,fr}} / \sim^\varepsilon$ рівна мірі простору $\vartheta_{(\vartheta_P|N_\chi)_{d_r,fr}} / \sim$.

Доведення.

◁

З адитивності міри випливає, що

$$\mu\left(\vartheta_{(\vartheta_P|N_\chi)_{d_r,fr}} / \sim^\varepsilon\right) = \\ = \mu\left(\vartheta_{(\vartheta_P|N_\chi)_{d_r,fr}} / \sim\right) + \mu\left(\vartheta_0(\vartheta_P|N_\chi)_{d_r,fr} / \sim^\varepsilon\right). \quad (46)$$

Оскільки за (45) $\mu\left(\vartheta_0(\vartheta_P|N_\chi)_{d_r,fr} / \sim^\varepsilon\right) = 0$, то

$$\mu\left(\vartheta_{(\vartheta_P|N_\chi)_{d_r,fr}} / \sim^\varepsilon\right) = \mu\left(\vartheta_{(\vartheta_P|N_\chi)_{d_r,fr}} / \sim\right). \quad (47)$$

▷

З (44) та (47) отримуємо обмеженість міри майже фактор-простору

$$\mu\left(\vartheta_{(\vartheta_P|N_\chi)_{d_r,fr}} / \sim^\varepsilon\right) \leq \mu(\vartheta_P). \quad (48)$$

Означення 2.

Змінну ε з (45) називатимемо довжиною майже факторизації.

Твердження 6.

Для ε має місце

$$\lim_{\varepsilon \rightarrow \infty} \mu\left(\vartheta_{(\vartheta_P|N_\chi)_{d_r,fr}} / \sim^\varepsilon\right) = \mu(\vartheta_P). \quad (49)$$

Доведення.

◁

Послідовність $\left\{ \mu\left(\vartheta_{(\vartheta_P|N_\chi)_{d_r,fr}} / \sim^\varepsilon\right) \right\}_\varepsilon$ є зростаючою, що випливає з адитивності міри.

З (48) випливає обмеженість цієї послідовності.

Отже, $\left\{ \mu\left(\vartheta_{(\vartheta_P|N_\chi)_{d_r,fr}} / \sim^\varepsilon\right) \right\}_\varepsilon$ є монотонною обмеженою послідовністю, а для такої послідовності за [] границя існує.

За оцінкою (36) максимум метрики (30) припадає при нульовій кореляції між фрагментами ϑ_P . Це означає, що вже при $\varepsilon = 1$ усі фрагменти ϑ_P потрапляють в $\vartheta_{(\vartheta_P|N_\chi)_{d_r,fr}} / \sim^\varepsilon$. Звідси при $\varepsilon \geq 1$ маємо

$$\vartheta_{(\vartheta_P|N_\chi)_{d_r,fr}} / \sim^\varepsilon = \vartheta_P \Rightarrow \mu\left(\vartheta_{(\vartheta_P|N_\chi)_{d_r,fr}} / \sim^\varepsilon\right) = \mu(\vartheta_P). \quad (50)$$

Отже, (48) виконується.

▷

Формула (49) не є суперечністю стосовно означення 2, оскільки майже усі (за винятком незначної кількості, якою можна знехтувати) фрагменти ϑ_P є еквівалентними за метрикою (30) заданому фрагменту. І якщо решта фрагментів таких, що задовольняють умову (44), утворюють множину міри нуль, то матиме місце (48).

Подібно до задачі 1 сформулюємо задачу майже факторизації:

Задача 3.

Майже факторизація простору $(\vartheta_P | N_\chi)$ полягає у побудові фактор-простору $\vartheta_{(\vartheta_P|N_\chi)_{d_r,fr}} / \sim^\varepsilon$ за допомогою метрики (30) і розширення цієї множини за рахунок множини міри 0, яка формується за (44).

Майже фактор-простір треба розглядати як логічний розвиток класичних фактор-просторів. Цілком природнім є розгляд ε -факторпростору як продовження (на величину ε) факторпростору $\vartheta_{(\vartheta_P|N_\chi)_{d_r,fr}} / \sim$. Але лише в тому випадку, коли обидва існують. В загальному випадку ε -факторпростір може існувати і без існуючого фактор простору $\vartheta_{(\vartheta_P|N_\chi)_{d_r,fr}} / \sim$. Але в цьому випадку за (46) його міра буде рівна 0. Цілком прийнятною є і зворотна ситуація: $\vartheta_{(\vartheta_P|N_\chi)_{d_r,fr}} / \sim$ – існує, але не визначено $\vartheta_0(\vartheta_P|N_\chi)_{d_r,fr} / \sim^\varepsilon$ – множини міри 0, для якої виконується (44). Тоді

$$\vartheta_{(\vartheta_P|N_\chi)_{d_r,fr}} / \sim^\varepsilon = \vartheta_{(\vartheta_P|N_\chi)_{d_r,fr}} / \sim. \quad (51)$$

Співвідношення (51) має місце тоді, коли розмірність множини $\vartheta_0(\vartheta_P|N_\chi)_{d_r,fr} / \sim^\varepsilon$ рівна нулю: $N_{\chi_0} / \sim^\varepsilon = 0$. В загальному випадку, незважаючи на твердження 3.10 розмірності фактор простору і

майже фактор простору можуть не збігатись. Це виражається рівністю

$$N_{\chi/\sim^\epsilon} = N_{\chi/\sim} + N_{\chi_0/\sim^\epsilon} . \quad (52)$$

У випадку набору \mathbf{P} (27) із заданою топологією $\mathcal{U}_{\mathbf{P}}$ (26) і фрагментним покриттям $(\mathfrak{P}_{\mathbf{P}} | N)$ задачу майже факторизації простору $(\mathfrak{P}_{\mathbf{P}} | N)$ можна сформулювати подібно до задач 2 і 3

Задача 4.

Майже факторизація простору $(\mathfrak{P}_{\mathbf{P}} | N)$ топології $\mathcal{U}_{\mathbf{P}}$ полягає у побудові фактор-простору $\mathfrak{P}_{(\mathfrak{P}_{\mathbf{P}} | N), d_r, \text{fr}} / \sim^\epsilon$ за допомогою метрики (30) і розширення цієї множини за рахунок множини міри 0, яка формується за (44).

Висновки

На прикладі метрики, побудованої на основі кореляції між фрагментами топологічного покриття зображення, сформульовано задачу факторизації топологічного покриття рисунка. Ця задача є базовою для дуже значної кількості практичних методів обробки зображень.

Із введенням інших напівметрич як відношень еквівалентності, можна сформулювати інші задачі факторизації і розробляти нові методи обробки.

Ще одним теоретичним результатом є введення поняття майже фактор-простору як операції, яка дає можливість формувати фактор-простори в межах незначних допущень.

Список літератури: 1. Александров, П. С. Введение в теорию множеств и общую топологию: учеб. пособие [Текст] / П. С. Александров. — М.: Наука, 1977. — 368 с. 2. Гонсалес, Р. Цифровая обработка изображений: пер. с англ. под ред. П. А. Чочиа [Текст] / Р. Гонсалес, Р. Вудс. — Москва: Техносфера, 2005. — 1072 с. 3. Колмогоров, А.Н. Элементы теории функций и функционального анализа: учеб. пособие [Текст] / А.Н. Колмогоров, С.В.Фомин. — 4-е изд. М.: Наука, 1976. — 544 с. 4. Колмогоров, А.Н. Основные понятия теории вероятностей: учеб. пособие [Текст] / А.Н.

Колмогоров. — 2-е изд. — М.: Наука, 1974. — 544 с. 5. Ленг, С. Введение в теорию дифференцируемых многообразий: пер. с англ. под ред. М.Я. Антоновского [Текст]. — М.: Мир, 1967. — 203 с. 6. Милнор Дж. Дифференциальная топология / Милнор Дж., Уоллес А., пер. с англ. под ред. Д.В.Аносова. — М.: Мир, 1972. — 280 с. 7. Соловьев, А.А. Лекции по теории вероятностей и математической статистике: учеб. пособие [Текст] / А.А.Соловьев. — Челябинск: Челябинский гос. ун-т, 2003. — 118 с. 8. Спивак, М. Математический анализ на многообразиях: пер. с англ. под ред. Д.А. Райкова [Текст] / М. Спивак. — М.: Мир, 1968. — 164 с. 9. Халмош, П. Гильбертово пространство в задачах: пер. с англ. под ред. Р.А. Милноса [Текст] / П. Халмош. — М.: Мир, 1970. — 352 с.

Надійшла до редколегії 31.03.2010 р.

УДК 681.142.2; 622.02.658.284; 621.325

Задача факторизации топологических пространств, определенных на изображениях / Д.Д. Пелешко, Ю.М. Рашкевич // Бионика интеллекта: науч.-техн. журнал. 2010. — № 1 (72). — С. 98–103.

В статье на примере введения метрики, построенной на основе статистической корреляции, рассматривается задача факторизации пространства, полученного из топологии изображения. Введено понятие почти факторизации с целью построения фактор-пространств при некоторых несущественных отклонениях отношения эквивалентности, полученного из заданой метрики.

Библиогр.: 10 назв.

УДК 681.142.2; 622.02.658.284; 621.325

The task of factorization of topological spaces determined accordingly to images / D.D. Peleshko, Y.M. Raskevych // Bionics of Intelligence: Sci. Mag. — 2010. — № 1 (72). — P. 98–103.

In the article based on the example of the introduction of the metric, built on the basis of statistical correlation, is described the task of the factorization of the space obtained from the image topology. Is introduced the concept almost of factorization for the purpose of construction the factor of spaces with some unessential deviations of the relation of equivalence, obtained from the defined metrics.

Ref.: 10 items.

УДК 004



ИНТЕЛЛЕКТУАЛЬНЫЕ АГЕНТЫ ДЛЯ СОЗДАНИЯ ПСЕВДО КОНТЕНТА ВЕБ-САЙТОВ

М.В. Збитнева

ХНУРЭ, г. Харьков, Украина, e-mail mayazbt@yandex.ru.

В настоящий момент популярным направлением является автоматическая генерация естественно-языковых текстов, результаты которой находят все большее применение в различных областях. Приведен анализ способов создания уникального контента, построены и расширены марковские модели, построен метод формирования псевдо-контента, определена архитектура интеллектуального агента.

ИНТЕЛЛЕКТУАЛЬНЫЙ АГЕНТ, МАРКОВСКИЕ МОДЕЛИ, ГЕНЕРАЦИЯ КОНТЕНТА, СОЗДАНИЕ КОНТЕНТА, ВЕБ-САЙТ

Введение

В настоящее время одним из способов, влияющих на скорость заработка при помощи Web, является использование дорвеев [1, 2]. Волнует вопрос, сколько сайт будет находиться в поисковой системе и сколько на этом можно будет заработать. Дорвей (doorway) – это набор веб-страниц, созданных под конкретный поисковый запрос, для поддержания высокого рейтинга в поисковых системах. В дорвеех используется большое количество ключевых слов и учитываются различные факторы, влияющие на ранжирование. Формирование дорвеев происходит на основе множества ключевых слов и текста тематических статей, сайтов. Пользователи, попавшие на такой сайт, часто перенаправляются на другой сайт. Одним из ключевых моментов в разработке таких сайтов является генерация контента.

В работе [3] представлено описание методов реферирования. В работе [4] идет речь о генерации кода. В работе [5] – вопросы синтеза текстов на естественном языке. Целью данного исследования является анализ способов генерации контента. В качестве задачи исследования выбрано определение архитектуры интеллектуального агента для автоматического создания псевдо-контента веб-сайта, построение марковской модели и создание метода по данной модели.

Сайт состоит из контента и программного кода. Контент – это та часть, которую практически невозможно защитить. Существующие способы защиты контента обеспечивают маленькую вероятность. К ним относятся:

- публикация в открытой печати;
- регистрация авторских прав;
- цифровые водяные знаки;
- использование средств отслеживания плагиата.

Контент представляет собой информационное наполнение сайта: текст, графические, аудио и видео файлы. Далее под контентом будет пониматься только текстовая масса.

Содержимым дорвеев является псевдо контент. Создание псевдо контента подразумевает автоматическое создание контента сайта на естественном языке.

Наряду с дорвеем широкое применение нашел клоакинг. Клоакинг – это методика отображения роботом поисковой системы оптимизированной страницы, в отличие от страницы, которую увидит обычный пользователь.

1. Клоакинг

Клоакинг реализуется с помощью программ и скриптов, выполняющихся на стороне веб-сервера. Серверные скрипты формируют выходные данные в зависимости от изменяющихся параметров: параметров в адресе запроса, доступных системных переменных веб-сервера и окружения, некоторых параметров запроса. По ряду данных можно определить от кого исходит запрос – от робота или обычного пользователя и сформировать для каждого отдельную результирующую страницу.

Существует два основных метода клоакинга:

- метод определения IP;
- метод определения UserName.

Метод определения IP. IP адрес идентифицирует подключение к Интернет. Пауки поисковых систем также имеют свои IP. Причем каждый паук имеет свой фиксированный IP адрес. Метод сводится к тому, что просто нужно определить IP посетителя, сравнить этот IP с базой данных (в которой записаны IP пауков поисковых систем) и сделать вывод посетитель это или паук. Если это посетитель, то показываем ему нормальную страницу. Если это паук - показываем ему оптимизированную страницу. Причем при создании оптимизированной страницы не нужно оглядываться на благозвучие текста, дизайн и тому подобное, потому что никто, кроме паука, эту страницу никогда не увидит.

Недостатки: Нужно иметь большую базу данных IP пауков, которую нужно периодически обновлять. Это достаточно дорогое удовольствие.

Преимущества: При хорошей базе данных вероятность обнаружения того, что используется клоакинг, крайне мала. Следовательно, методы оптимизации останутся не известными.

Метод определения UserName. Суть этого метода та же, что и предыдущего. Только определять нужно не IP, а UserName. Последовательность

действий такая же: определяем, сравниваем, показываем нужную страницу.

Недостатки: `UserName` можно подделать и “прикинуться” поисковым пауком. И таким образом узнать все секреты оптимизации.

Преимущества: Можно без проблем получить базу данных `UserName` пауков. То есть простота реализации.

Оптимальное решение

Самый надежный способ реализации это совмещение двух методов. То есть определение и IP и `UserName`. И чтобы не было проблем, в случае если даже IP сходится, а `UserName` - нет, то показывать нормальную страницу.

Таким образом, одна из сфер применения генерации контента – это использование клоакинга со стороны сервера, поскольку робот не учитывает качество текстовой массы.

2. Способы создания уникального контента

Способы создания уникального контента:

– написанный вручную:

а) копирайтинг – создание уникального текста с целью продвижения на рынке товаров и услуг; самостоятельное написание статей для сайтов, блогов; покупка готового уникального контента на текстовых биржах;

б) рерайтинг – переписывание статьи своими словами с сохранением смысла для размещения на сайтах, публикация в СМИ;

в) перевод контента.

– сгенерированный контент:

а) копипастинг:

– составление шаблона предложений;

– заполнение шаблона из множества контекста.

б) уникализация текста:

– добавление опечаток;

– замена русских букв на латинские;

– смена предложений и абзацев местами.

в) синонимизация:

– подбор синонимов;

– перемещение слов в предложении;

– перемещение предложений.

г) компьютерная лингвистика.

Существуют также способы автоматической генерации форматирования текста и программного кода [4].

3. Способы автоматической генерации текста

Автоматическая генерация текста предполагает использование программного обеспечения, которое создает последовательности символов, как правило, лишённые смысла; позволяет создать быстро и уникально по правилам русского языка текст на заданную тему. Сгенерированный текст содержит существительные, прилагательные, глаголы и наречия.

При этом соблюдается естественное разнообразие и грамматически правильное согласование

падежей, лиц, чисел, времен, форм, родов и так далее. Тексты могут проходить проверку орфографии в Word. Наличествуют различные знаки препинания, предлоги и союзы. Все предложения начинаются только с заглавной буквы, присутствуют сложные предложения.

Присутствует обработка слов по сочетаемости – между существительными, существительными с глаголами, существительными с прилагательными и глаголов с наречиями.

При генерации текста на определенную тему подбор слов осуществляется так, что слова в данной области будут встречаться намного чаще, чем остальные.

При автоматической генерации сложно добиться естественного языка, можно получить грамматически правильно читаемый текст без особого смысла, насыщенный разнообразными словами, иногда со смешными фразами и целыми фрагментами текста.

Автоматическая генерация текста также является одним из направлений компьютерной лингвистики [5], наряду с анализом текста, пониманием текстов, оживлением текстов, моделями коммуникации. Для синтеза текста могут использоваться актанты действий. С каждым действием связан набор объектов и характеристик. Например с действием «ехать» связаны субъект, который совершает движение, начало и конец движения и так далее. Все эти данные объединяются в структуры – фреймы. Таким образом, процесс синтеза текста выглядит следующим образом:

– генерация нужной последовательности глаголов;

– заполнение актантных структур, создание глубинной структуры предложений;

– связь структур по общим действующим субъектам и объектам;

– создание синтаксически правильных структур предложений.

Одной из областей применения являются системы автоматического реферирования [3], которые формируют краткое изложение либо путем выделения фрагментов информационного наполнения и последующего их соединения, либо методом генерации текста на основании знаний об оригинале. Здесь используется метод линейных весовых коэффициентов. Веса назначаются для блока текста: учитывают расположение блока в оригинале, частоту появления в тексте, частоту использования в ключевых предложениях, показатели статистической значимости. Весовой коэффициент расположения зависит от начала, конца, середины текста, его присутствия в ключевых разделах: введение, заключение. Статистическая важность вычисляется на основании данных, полученных в результате анализа автоматической индексации. Здесь учитывается появление термина в заголовке, колонтиту-

ле, первом параграфе, пользовательском профиле запроса. Данный метод не учитывает связи между предложениями. В ряде подходов создается специальное окно для предыдущего предложения реферата, с помощью которого можно определить наличие смыслового разрыва или «висящего» слова. В других случаях предложения, содержащие «висящие» слова, исключаются из реферата, либо делаются попытки разрешения ссылок, которые предполагают такие слова или их сверки путем краткого лингвистического анализа. При таком подходе степень сжатия уменьшается, так как в реферат привносится посторонняя информация. Кроме того, когда основной реферат уже сформирован, трудно восстановить исходный процент сжатия.

Следующий метод основан на подборе выдержек. Здесь необходимы грамматики, словари синтаксического разбора, онтологии. Данный метод использует традиционный лингвистический метод синтаксического разбора предложений. В этом методе применяется семантическая информация для аннотирования деревьев разбора.

Процедуры сравнения манипулируют непосредственно деревьями с целью удаления и перегруппировки частей, например путем сокращения ветвей на основании некоторых структурных критериев, таких как скобки или встроенные условные или подчиненные предложения. После такой процедуры дерево разбора существенно упрощается, становясь, по существу, структурной «выжимкой» исходного текста.

Данный подход относится к системам искусственного интеллекта и опирается на понимание естественного языка. Синтаксический разбор также входит составной частью в такой метод анализа, но деревья разбора в этом случае не порождаются. Напротив, формируются концептуальные репрезентативные структуры всей исходной информации, которые аккумулируются в текстовой базе знаний. В качестве структур могут быть использованы формулы логики предикатов или такие представления, как семантическая сеть или набор фреймов. Примером может служить шаблон банковских транзакций (заранее определенное событие), в котором перечисляются организации и лица, принимающие в нем участие, дата, объем перечисляемых средств, тип транзакции и так далее.

4. Интеллектуальные агенты для создания псевдо контента веб-сайтов

Одним из распространенных способов создания динамических приложений, меняющих свое поведение или состояние в режиме реального времени, являются интеллектуальные агенты. Из наиболее известных архитектур интеллектуальных агентов [6] для данной задачи выбрана архитектура агента, основанного на модели. Модель представляет собой расширенную марковскую модель, которая

описывается ниже. Марковская модель широко используется для автоматической генерации текста. Основное применение метода автоматического создания контента – генерация дорвейного бизнеса.

При генерации текста самый простой способ выбора следующей буквы из множества возможных следующих букв – это равновероятный выбор любой буквы. Второй – это подсчет частоты встречаемости букв в языке. Третий – цепи Маркова, где исследуются не отдельные буквы и слова, а их пары. Цепи Маркова показали себя на практике как один из наиболее эффективных способов для решения задачи генерации текста.

Способы генерации цепями Маркова:

Берется определенный текст. Выбирается N последовательных слов из текста, которые будут являться первыми сгенерированными словами случайного текста. Далее каждое последующее слово определяется следующим образом. Берутся последние N сгенерированных слов и рассматриваются все вхождения этой фразы в исходный текст. Рассматриваются все слова, которые идут в этом тексте сразу после этих вхождений, и из них выбирается случайное – оно и будет следующим словом сгенерированного текста.

В качестве текстовой массы можно использовать результаты поиска серверов. В качестве первой фразы – фразу для поиска. В качестве продолжения выбирается случайная фраза из результатов поискового сервера.

Для формирования слов могут использоваться частотные характеристики средней длины слова и длины предложения.

Рассмотрим пример 1 для фразы «Марковские модели для создания псевдо контента веб-сайтов». Частота встречаемости для нее приведена в табл. 1.

Таблица 1
Частота встречаемости для примера 1

Первая буква	Следующая буква (частота встречаемости)					
	р(1)	й(1)	н(1)	пробел(1)		
А						
О	в (2)	д(1)	з(1)	пробел	н(1)	
И	е(1)	пробел	я(1)			
Е	пробел	л(1)	в(1)	н(1)	б(1)	
Я	пробел(2)					
В	с(1)	д(1)	е(1)			
М	а(1)	о(1)				
Р	к(1)					
К	о (2)	и(1)				
С	к(1)	о(1)	е(1)	а(1)		
Д	е(1)	л(1)	а(1)	о(1)		
Л	и(1)	я(1)				
З	д(1)					
Н	и(1)	т(2)				
П	с(1)					
Т	е(1)	а(1)	о(1)			
Б	дефис					
Й	т(1)					
Пробел	м	д	с	п	к	в
Дефис						

Для буквы «А» существует следующая ей последовательность (р, й, н, пробел). Для буквы «В» – (с, д, е). Для буквы «О» – (в, д, з, пробел, н, в) и так далее. То есть для буквы «О» буква «в» встречается в два раза чаще, чем остальные «д, з, пробел, н» буквы.

Построим марковскую модель, которая в зависимости от контекста буквы может описывать, слова, предложения, абзацы, называемый каждый в отдельности блоком текста.

Матрица вероятностей переходов имеет вид:

$$P=[p_{i,j}] = \begin{matrix} & S_1 & \dots & \dots & S_n \\ S_1 & P_{1,1} & \dots & \dots & P_{1,n} \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ S_n & P_{n,1} & \dots & \dots & P_{n,n} \end{matrix},$$

где $p_{i,j}$ – вероятность появления j -го блока текста после i -го.

Значения матрицы могут принимать значения в интервале от нуля до единицы. Значение «ноль» соответствует отсутствию такой комбинации в оригинале.

Множество состояний описывается множеством $S = \{S_1, \dots, S_n\}$. Мощност множества соответствует количеству уникальных блоков текстовой массы. Мощност множества соответствует количеству уникальных пар блоков текстовой массы. Граф состояний показан на рис. 1.

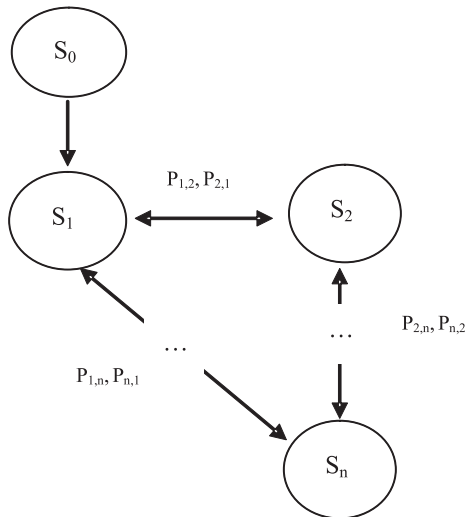


Рис. 1. Граф марковской цепи

При формировании моделей вектор начальных вероятностей выглядит как $\{1, 0, \dots, 0\}$. Стартовой точкой является начало текстовой массы. При формировании текстовой массы по модели стартовой точкой может являться любое состояние множества S , а именно состояние, содержащее ключевые блоки текста. В случае асинхронного, параллельного процесса начальных состояний может быть несколько.

Марковская модель может моделировать построение слова, словосочетания, предложения, абза-

ца и целого текста как отдельно, так и в совокупности, представляя собой большой массив данных. С матрицей вероятностей переходов может быть связана матрица атрибутов, которая содержит различные семантические и синтаксические характеристики блоков текста и их взаимосвязей, их онтологическое представление в реальном мире, то есть база знаний.

Таким образом, метод генерации псевдо – контента заключается в следующем:

1) Построение марковской модели блоков текстовой массы ArrayText.

$\forall \text{Connect}(X, Y) \Rightarrow$
 $\Rightarrow \text{PopulateArrayText}(X, Y, \text{Connect}(X, Y))$,
 где X, Y – блоки текстовой массы.

2) Формирование матриц атрибутов ArrayAttributeRelation (X, Y) , ArrayAttributeUnitText (Z) , где Z – блок текстовой массы.

3) Сканирование марковской модели, последовательно идя со стартовой точки, формируя слова, предложения, абзацы до указанной длины.

BuildUnitText (LengthText, ArrayText, ArrayAttribute) = $\exists \text{ArrayText}(X, Y) \wedge$
 $\wedge \exists \text{ArrayAttribute}(X, Y) \wedge \exists \text{LengthText}$.

4) Обвертывание полученной текстовой массы форматом для отображения в веб.

BuildWebTemplate = AddWebTemplate (BuildUnitText (LengthText, ArrayText, ArrayAttribute)).

Для формализации использована логика предикатов первого порядка.

Рассмотрим пример 2. Построим марковскую модель для слова «контент». Множество состояний $S = \{к, о, н, т, е\}$. Вектор начальных вероятностей $\pi = \{1, 0, \dots, 0\}$. Граф состояний для примера имеет вид, показанный на рис. 2.

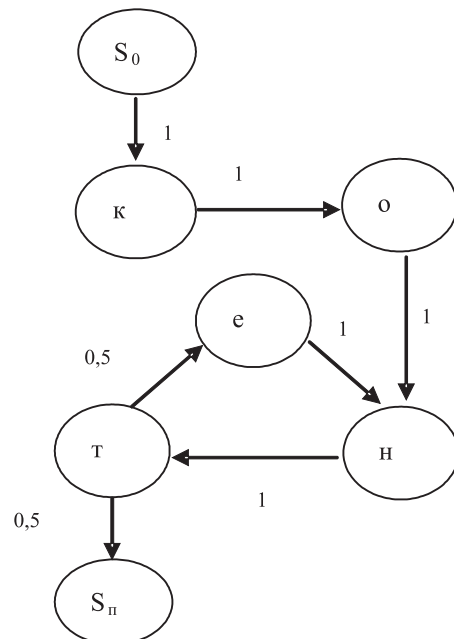


Рис. 2. Граф марковской цепи для примера 2

Матрица вероятностей переходов:

		к	о	н	т	е	н	т
P=[p _{ij}]=	к	0	1	0	0	0	0	0
	о	0	0	1	0	0	0	0
	н	0	0	0	1	0	0	0
	т	0	0	0	0	0,5	0	0
	е	0	0	0	0	0	1	0
	н	0	0	0	0	0	0	1
	т	0	0	0	0	0	0	0,5

Выводы

Таким образом, в результате проведенного исследования проанализированы существующие способы генерации контента, усовершенствована марковская модель, описывающая зависимость блоков текста, предложен метод формирования блоков псевдо текста, использующий предложенную модель.

Полученные результаты применяются для формирования текстов-дорвеев. К перспективе дальнейшего исследования относится исследование семантики сгенерированных текстов.

Список литературы: 1. *Нартова, А.* Прописка в Интернете [Текст] / А. Нартова, А. Набережный // Мир ПК. 2002. – №12. – С. 5-9. 2. *Керк, Д.* «Вредный» поиск. [Текст] / Д. Керк // ComputerWorld. – 2007. – №15. – С. 37. 3. *Хан, У.* Системы автоматического реферирования [Текст] / Удо Хан, Индерджиет Мани // Открытые системы. – 2000. – № 12. – С. 18-22. 4. *Корвяченко, А.* Алгоритмы кодогенерации. Кодогенерация при программировании

с использованием платформы Microsoft.NET. [Текст] / А. Корвяченко // RSDN Magazine. – 2003. – № 4. – С. 16-24. 5. *Коробова, И.Л.* Автоматизированная система синтеза текста на основе технологии шаблонизации [Электронный ресурс] / ГОУ ВПО «Тамбовский государственный технический университет «Национальный фонд подготовки кадров Педагогический Интернет-клуб Тамбовской области // Материалы межрегион. науч.-практ. конф. «Информатизация системы образования Тамбовского региона». – Режим доступа : <http://club-edu.tambov.ru/main/news/index.php?r=konfl&f=t12>, свободный. 6. *Рассел, С.* Искусственный интеллект: современный подход, 2-е изд. [Текст] / С. Рассел, П. Норвиг. – Москва: Изд. дом «Вильямс», 2006. – 1408 с.

Поступила в редколлегию 06.04.2010 г.

УДК 004

Марківські моделі для створення псевдо-контенту веб-сайтів / М.В. Збітнева // Біоніка інтелекту: наук.-техн. журнал. – 2010. – № 1 (72). – С. 104–108.

Генерація контенту знаходить своє призначення у різних галузях. В статті проаналізовано способи генерації контенту, наведено марківські моделі, архітектуру інтелектуального агенту, а також метод формування веб-контенту.

Табл. 1. Іл. 2. Бібліогр.: 6. найм.

UDK 004

Markov models for web-site content pseudo-generation. / M.V. Zbitneva // Bionics of Intelligence: Sci. Mag. – 2010. – № 1 (72). – P. 104–108.

Content generation is used in different domain. There are analyzed ways of content generation, presented markov models, agent architecture, and method forming web-content.

Tab. 1. Fig. 2. Ref.: 6. items.

УДК 004.942, 004.891.2



МЕТОДИ ФРАКТАЛЬНОЇ ГЕОМЕТРІЇ ТА НЕЧІТКОЇ ЛОГІКИ, ЯК ОСНОВА МАТЕМАТИЧНОГО МОДЕЛЮВАННЯ ПРОСТОРОВОЇ ОРГАНІЗАЦІЇ ТУРИСТИЧНИХ ПОСЕЛЕНЬ

Я.І. Вихлюк

НУ «Львівська політехніка», м.Львів, Україна, vyklyuk@ukr.net

Проведено аналіз досліджень у сфері фрактального моделювання урбанізованих територій. Розроблена методика моделювання форми та структури туристичних поселень на основі самоафінних фракталів у нечіткому імовірнісному полі.

САМОАФІННИЙ ФРАКТАЛ, НЕЧІТКА ЛОГІКА, РЕКУРСІЯ, ФРАКТАЛЬНА РОЗМІРНІСТЬ, УРБАНІЗАЦІЯ, ГІС

Вступ

Стрімкий розвиток туристичної галузі вимагає використання швидких, кількісних, адекватних математичних методів та методик при прийнятті рішень. Одною з найбільш актуальних задач туристичної галузі є прогноз структури та форми туристичних поселень та урбанізованих територій. Низький рівень адекватності класичних математичних моделей ставить перед науковцями питання про розробку та впровадження сучасних методів математичного моделювання, таких як SoftComputing та фрактальна геометрія.

1. Аналіз дослідження у сфері моделювання просторової організації міст

Метою наукових досліджень просторової організації було вивести ідеалізовану теорію розбудови міст у рамках жорстких обмежень. Довгий час просторова організація міст описувалась за допомогою геометрії Евкліда. Однак ідеалізовані міста були далекі від реально існуючих. Основною проблемою була саме неможливість застосування класичної математики до реальних об'єктів. За допомогою геометрії Евкліда можна виміряти такі фундаментальні величини, як довжина, площа, тощо. Однак в реальних містах базовою «цеглиною» у більшості випадків є одиночний будинок. Будинки в свою чергу утворюють квартали. Кwartали складаються з будинків, які впорядковані за певними законами самоподібності та обмежені дорогами, що відділяють квартали один від одного. Кwartали утворюють самоподібну структуру, що залежить від спеціалізації атракторів, навколо яких ведеться забудова міста. У якості атракторів може виступати завод, розважальний центр, церква, ринкова площа міста, тощо [1].

Згідно з геометрією Евкліда ідеальні міста мають характеризуватись регулярністю. В той самий час, переважна більшість міст є нерегулярною та може бути описана геометрією Мандельброта [2]. У першому випадку міста мають мати сферичну форму. Насправді на форму міст впливають транспортні шляхи, що деформують сферу вздовж транспорт-

них артерій [3]. Класично міста представляються у вигляді абсолютної регулярної дискретної сітки основних типів поселення, що систематично урбанізовані та утворюють міста, села та регіони. Насправді поселення характеризуються неперервною структурою та можуть займати нецілу кількість клітинок або розташовуватись на гранях сітки. Аналогічно виникає проблема з визначенням периметру населеного пункту. При будь-якому збільшенні масштабу вимірювання з'являються нові нерівності в структурі. У граничному випадку периметр буде прямувати до безмежності. Як було показано в роботі [4], такі структури володіють нецілою розмірністю Хаусдорфа-Базікевича, а самі системи можуть бути змодельовані за допомогою фрактальної геометрії. Нормою в містобудівних системах є представлення міста у вигляді вулиць та будинків. Часто при будівництві нових житлових масивів приймається рішення про будівництво паралельних та прямих вулиць. Міста, які мають багатовікову історію та розвивались без генерального планування, не містять паралельних вулиць, чистої симетрії і на перший погляд володіють хаотичною структурою. Однак ця хаотична та самоподібна структура утворює гармонійне місто. Можна зробити висновок, що нерегулярність та нерівномірність форм сучасного міста є нормою, а не винятком.

Для застосування фрактальної геометрії введемо поняття населений пункт: це множина різних базових елементів та об'єднання їх у підсистеми, що відображають базові форми на різних рівнях ієрархії. Дана множина організована таким чином, що в будь-яких просторових вимірах є повторенням базових елементів у різних масштабах. Дане формулювання дає підстави стверджувати, що міста мають фрактальну форму. Отже, моделювання міст необхідно проводити за принципом ієрархії. А форма міст визначається зовнішніми чинниками, що виступають у ролі обмежень (водойма, аграрні землі, залізниця, кордон, тощо) або у ролі поля імовірності урбанізації.

Моделювання периферії населених пунктів, як правило, проводиться методами дифузно-обме-

женої агрегації, або клітинними апаратами. Достатньо відомим методом моделювання урбанізації території є модель клітинної урбанізації (Cellular Urban Model)[5]. Для визначення імовірнісних правил неперервної дифузії частинок використовуються навчальні множини, отримані за допомогою гео-інформаційних систем (ГІС). Врахування нелінійності проводиться за допомогою методу опорних векторів. Запропонований підхід був апробований при моделюванні розвитку міста Шензен, Китай, та показав високу точність. В праці [6] проводиться розрахунок імовірнісного поля за допомогою математичних методів, що застосовуються при обробці зображень. Моделі були апробовані при моделюванні розвитку гірського містечка Iwaki Newtown та показали високий рівень адекватності. Використання Cellular Urban Model до моделювання двох різних столичних районів у Португалії [7] дозволило провести порівняльний аналіз динаміки розвитку досліджуваних регіонів. В статті [8] вдосконалюється апарат Cellular Urban Model в рамках формалізму ГЕО-алгебри та представлений механізм інтеграції між даними ГІС та традиційними класами міських і регіональних моделей. Детальний опис математичного апарату Cellular Urban Model наводиться в дослідженні [9]. Тут же приводиться моделювання розвитку міста Сага, Японія.

Дані підходи дозволяють отримати непогане узгодження форми периферії з експериментом, однак втрачається інформація про внутрішню структуру міста.

Метою даної роботи є розробка методики моделювання форми населених пунктів, що спеціалізуються на туристичних послугах, методами математичних самоафінних фракталів. Це дозволить без втрати інформації про внутрішню структуру за короткий час комплексно змодельовати структуру міста.

2. Самоафінні фрактали, як основа моделювання внутрішньої структури міст

В роботі [10] доводиться, що міста мають деревоподібну ієрархічну структуру. Це можна продемонструвати на розвитку містечка Savannah, Georgia (рис.1). Це рідкий випадок, коли на ріст населеного пункту впливає мало збурень, що приводить до

симетричного розвитку останнього вздовж річки.

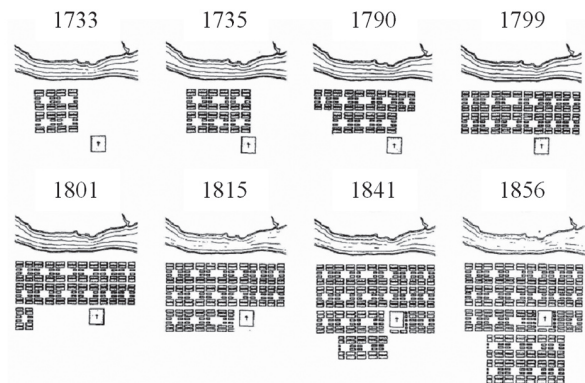


Рис. 1. Регулярний клітинний ріст населеного пункту Savannah, Georgia 1733 – 1856 pp. [10]

Особливості зростання такого типу містечок можуть бути змодельовані процесом фрактального росту, зображеного на рис. 2 [11]. Алгоритм побудови наступний. Спочатку задається ініціатор $\vec{V}(z_1, z_2)$, що представляє пряму лінію. Ініціатор це геометрична фігура, що є аналогом затравки у фізичних кристалах. Тобто об'єкт, навколо якого буде зростати фрактал. Аплікуючи на населені пункти, ініціатор це дорога, навколо якої розбудовується населений пункт. В якості координат вектора зручно використати комплексні числа, дійсна та уявна частина яких визначає географічну довготу та широту точки $z_i = x_i + iy_i$. На другому кроці алгоритма створюються дві копії ініціатора $(\vec{V}_1^1, \vec{V}_1^2)$ з масштабом S , над якими проводять наступні афінні перетворення: кожна з копій розвертається на кут $\pi/2$ та зміщується вздовж базового ініціатора на відстань $l = \frac{1}{2}(z_1 + z_2)$ в напрямку z_1 та z_2 відповідно. Математично дані афінні перетворення можна представити у вигляді [12]:

$$\vec{V}_1^1 = S \cdot e^{i\pi/2} \cdot \left(\vec{V}_0 - \frac{1}{2}(z_1 + z_2) \right) + z_1; \quad (1)$$

$$\vec{V}_1^2 = S \cdot e^{i\pi/2} \cdot \left(\vec{V}_0 - \frac{1}{2}(z_1 + z_2) \right) + z_2. \quad (2)$$

На третьому кроці кожний з отриманих векторів \vec{V}_1^1, \vec{V}_1^2 виступає в ролі ініціатора, і над ними проводяться аналогічні операції копіювання та афінні перетворення (1)-(2).

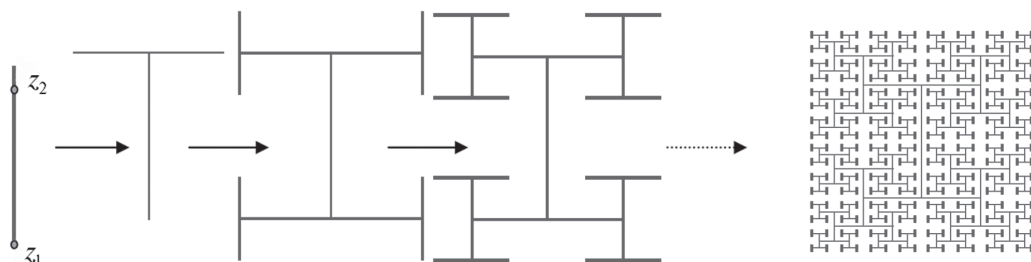


Рис. 2. Схема побудови фрактала, що моделює внутрішню структуру населеного пункту

Вектори останнього кроку рекурсії інтерпретуються як будинки, а попередніх – дороги. Дана рекурсія проводиться визначену парну кількість разів, так щоб фізичний розмір будинків був співрозмірним із середнім розміром будівель реального населеного пункту.

Необхідною умовою рівномірного заповнення площини будинками є рівність довжини векторів, отриманих на другому кроці ітерації (l_2), половині довжини ініціатора $l_0 = |\vec{V}|$. В ході афінних перетворень над ініціатором двічі здійснюється операція масштабування з масштабним множником S . Отже довжина l_2 визначається як :

$$l_2 = S^2 \cdot l_0 ; \quad (3)$$

тоді

$$S^2 \cdot l_0 = \frac{1}{2} l_0 \Rightarrow S = \frac{1}{\sqrt{2}} \approx 0,71 . \quad (4)$$

Якщо $S > 0,71$ лінії, що утворюють фрактал, перекриватимуться. При $S < 0,71$ фрактал почне групуватись у кластери («квартали») (рис. 3).

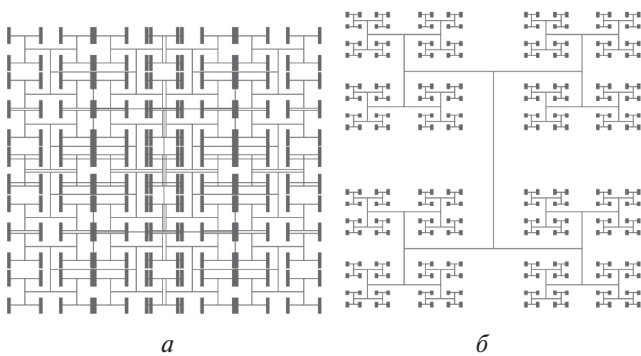


Рис. 3. Вигляд фракталу при $S = 0,78$, $D = 2,8$ (а) та $S = 0,64$, $D = 1,55$ (б)

Фрактальна розмірність такого самоафінного фракталу визначається так: на кожній наступній ітерації ініціатор ділиться на дві частини $N = 2$, з масштабним коефіцієнтом $S = \frac{1}{\sqrt{2}}$. Тоді фрактальна розмірність складає: $D = -\ln(N)/\ln(S) = 2$. Дійсно, при збільшенні ітерацій до безмежності такий самоафінний фрактал рівномірно заповнює квадратну область.

Іншим підходом до моделювання внутрішньої структури населених пунктів є фрактал дерева

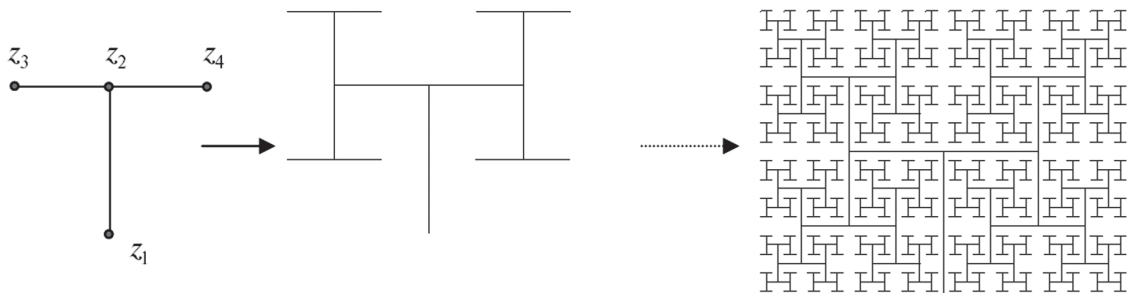


Рис. 4. Схема побудови дерева Піфагора

Піфагора, або модель людських легень, запропонована Мандельбротом [11]. Ініціатором даного самоафінного фрактала є три вектори, що визначають стебло дерева та дві гілки. Координати векторів зручно представити у вигляді масива $V = (z_1, z_2, z_3, z_4)$. Де z_i – координати вершин векторів (рис. 4): $\vec{V}_i(z_1, z_2)$ – стовбур ініціатора, $\vec{V}_l(z_2, z_3)$ – гілка ліворуч, $\vec{V}_r(z_2, z_4)$ – гілка праворуч. На першому кроці ітерації з кожного закінчення гілки колінеарно (паралельно та антипаралельно) ініціатору будуються по дві копії ініціатора в масштабі S . Дані афінні перетворення задаються виразами:

$$V_1^1 = S \cdot e^{i \cdot \xi} \cdot (V - z_1) + z_3 , \quad (5)$$

$$V_2^1 = S \cdot e^{i \cdot (\pi - \xi)} \cdot (V - z_1) + z_3 , \quad (6)$$

$$V_3^1 = S \cdot e^{i \cdot (\pi + \xi)} \cdot (V - z_1) + z_4 , \quad (7)$$

$$V_4^1 = S \cdot e^{-i \cdot \xi} \cdot (V - z_1) + z_4 , \quad (8)$$

де ξ – збурення, що дозволяє моделювати деформацію фракталу.

Аналогічні ітерації повторюються задану кількість разів. Гілки останньої ітерації розглядаються як будинки.

Для рівномірного заповнення площини фракталом гілки на першій ітерації мають бути вдвічі меншими від гілок ініціатора, отже масштабний множник в афінних перетвореннях $S = 0,5$. Фрактальна розмірність такого фракталу складає: $D = -\ln(4)/\ln(0,5) = 2$. Це підтверджує тезу, що даний фрактал рівномірно заповнює площину в граничному випадку, коли кількість рекурсій прямує до безмежності.

У випадку моделювання внутрішньої структури населеного пункту, що складається із сітки доріг, необхідно поділити вектори доріг на ініціатори та обмеження. В ролі ініціаторів виступають основні дороги, навколо яких історично почалась забудова. Обмеженнями слугують дороги, що обмежують квартали. Для кожного з ініціаторів будується самоафінний фрактал (1)-(2), ріст якого обмежується векторами доріг-обмежень.

Дещо інший підхід представлено в роботі [13]. Сітка доріг населеного пункту представляється як самоподібна деревовидна структура. Гілки остан-

нього рівня слугують ініціаторами, а гілки верхніх рівнів – обмеженнями (рис. 5). В такому представленні для моделювання внутрішньої структури використовується фрактал дерева Піфагора.

Дані підходи дозволяють моделювати внутрішню структуру населеного пункту. Для моделювання форми периферії необхідно ввести поле, що визначає імовірність урбанізації тієї чи іншої території.

3. Метод побудови поля імовірності урбанізації

При моделюванні просторового розвитку невеликих туристичних містечок необхідно виділити особливості їх розвитку, що стануть основою для вибору методу та алгоритму моделювання.

Перш за все форма периферії таких утворень розвивається самостійно, без генерального планування. Під формою будемо розуміти просторову структуру елементів, з яких складаються міста, до яких можна віднести інфраструктурні мережі, будівлі, урбанізовані території, що визначається через її геометрію виключно у двох, а не трьох вимірах. Відсутність генерального планування означає відсутність чітких границь населеного пункту. Адже периферію забудовує або місцеве населення, або малий та середній бізнес, що власноруч приймає рішення про викуп землі та розбудову. Тобто наявність чи відсутність забудови можна визначити, ввівши імовірність урбанізації.

По-друге, туристичні містечка орієнтовані на пішоходів. Це накладає обмеження на просторовий розмір населених пунктів. Адже розбудова таких поселень пов'язана з наданням послуг з проживання туристів, що приїхали заради відпочинку на атракторах до яких можна дістатись пішки. Збільшення розмірів поселень призводить до необхідності введення міського транспортного сполучення. Що вимагає використання інших підходів та моделей.

По-третє, на відміну від великих міст, що містять декілька атракторів з різною спеціалізацією, туристичні містечка орієнтовані на надання однотипних послуг, перелік яких диктуються рекреаційним ат-

рактором. Тому в останніх відсутня яскраво виражена сегментація на зони спеціалізації.

По-четверте, внутрішня структура туристичних містечок в основному зумовлена під'їзними шляхами, якими рухаються туристичні потоки. А дороги-обмеження кварталів містечок не варто брати у розрахунок, у зв'язку з тим, що це, як правило, ґрунтові дороги і вони можуть бути легко змінені за напрямком при умові розбудови. Тому при моделюванні треба враховувати не дороги-обмеження, а сфери впливу доріг ініціаторів.

Можна прийти до висновку, що основними факторами, які визначають привабливість території до забудови, є координати атракторів, відстань до останніх по основним транспортним шляхам та відстань по ґрунтовій дорозі до траси. Такі величини, як водойми, річки, залізниця, тощо виступають у ролі обмежень. Вищевказані вхідні параметри та обмеження можна отримати з ПС систем. Більшість з них містяться в даних системах явно, а деякі потребують додаткових розрахунків. Зокрема для визначення віддалі дорогою до атрактора та відстані до траси необхідно здійснити наступні перетворення.

Нехай дороги досліджуваної території задані масивом:

$$w_f(x_{f1}, y_{f1}, x_{f2}, y_{f2}), f = \overline{1, n}, \quad (9)$$

де n – кількість векторів доріг на досліджуваній території; $x_{f1}, y_{f1}, x_{f2}, y_{f2}$ – координати вектора дороги.

Тоді, віддаль h точки з координатами x, y до найближчої дороги визначається згідно з наступними міркуваннями: розглянемо трикутник з вершинами $A(x, y), B(x_{f1}, y_{f1}), C(x_{f2}, y_{f2})$ (рис. 6), довжина сторін якого визначається як:

$$a = \sqrt{(x_{f1} - x_{f2})^2 - (y_{f1} - y_{f2})^2}, \quad (10)$$

$$b = \sqrt{(x_{f1} - x)^2 - (y_{f1} - y)^2}, \quad (11)$$



а



б

Рис. 5. Зображення міста Coventry [13] (а), та спрощений план дорожньої системи (б)

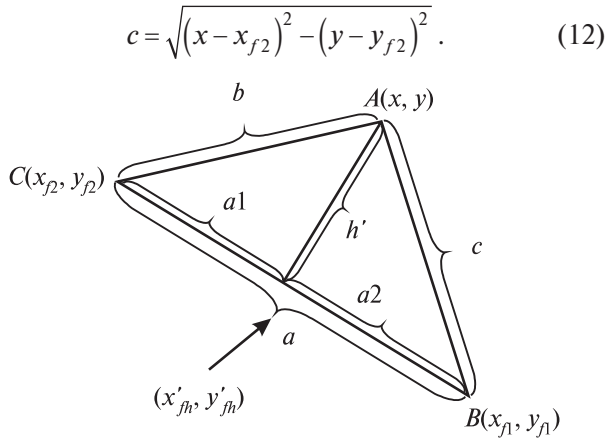


Рис. 6. Координати трикутника для визначення висоти h' та координат базиса висоти (x'_{fh}, y'_{fh})

Висота трикутника на пряму, якій належить сторона a (відрізок дороги):

$$h'_f = \frac{2\sqrt{p(p-a)(p-b)(p-c)}}{a}, \quad (13)$$

$$p = \frac{a+b+c}{2}. \quad (14)$$

Відрізки a_1 та a_2 , що визначають віддаль між основою висоти до вершин В та С відповідно, та координати основи висоти (x'_h, y'_h) визначаються наступним чином:

$$a_1 = \sqrt{b^2 - h'^2_f}, \quad (15)$$

$$a_2 = \sqrt{c^2 - h'^2_f}, \quad (16)$$

$$x'_{fh} = x_{f1} - (x_{f1} - x_{f2}) \frac{a_1}{a}, \quad (17)$$

$$y'_{fh} = y_{f1} - (y_{f1} - y_{f2}) \frac{a_1}{a}. \quad (18)$$

Висота (13) є найкоротшою віддаллю до дороги у випадку, якщо точка (x'_{fh}, y'_{fh}) лежить на відрізку дороги f . В іншому випадку, найкоротша віддаль до дороги визначатиметься:

$$h_f = \begin{cases} a_1 + a_2 = a, h'_f \\ a_1 + a_2 > a \text{ and } b < c, b \\ a_1 + a_2 > a \text{ and } b > c, c \end{cases} \quad (20)$$

Координати точки перетину відповідно:

$$(x_{fh}, y_{fh}) = \begin{cases} a_1 + a_2 = a, (x'_{fh}, y'_{fh}) \\ a_1 + a_2 > a \text{ and } b < c, (x_1, y_1) \\ a_1 + a_2 > a \text{ and } b > c, (x_2, y_2) \end{cases} \quad (21)$$

Тоді відстань до найближчої дороги визначається як:

$$h = \min_{f=1, n} (h_f). \quad (22)$$

Розрахунок довжини оптимального шляху від точки (x_{fh}, y_{fh}) до найближчого атрактору визначається шляхом розв'язання наступної оптиміза-

ційної задачі лінійного програмування. Спочатку масив w_f необхідно перетворити в масив, що містить всі можливі географічні координати масиву w_f : $v = \{x_j, y_j\}, j = \overline{1, m+2}$, де $(x_1, y_1) = (x_{fh}, y_{fh})$, $(x_{m+2}, y_{m+2}) = (x_{att}, y_{att})$, (x_{att}, y_{att}) – координати найближчого атрактора, x_j, y_j – географічні координати із масиву w_f , m – кількість різних географічних координат, які містить масив w_f .

На основі масиву v створюється симетрична матриця відстаней $A = \{a_{ij}\}, i, j = \overline{1, m+2}$, колонки та рядки якої відповідають географічним координатам із масиву v , a_{ij} дорівнює довжині вектора між відповідними координатами у випадку наявності прямого транспортного сполучення між ними і 0 в протилежному випадку:

$$a_{ij} = \begin{cases} \sqrt{(v[i,1] - v[j,1])^2 + (v[i,2] - v[j,2])^2}, \\ \{v[i,1], v[i,2], v[j,1], v[j,2]\} \in w \\ 0, \\ \{v[i,1], v[i,2], v[j,1], v[j,2]\} \notin w \end{cases} \quad (23)$$

У якості змінних рішення виступає двійкова матриця $B = \{b_{ij}\}, i, j = \overline{1, m+2}$, b_{ij} дорівнює 1 у випадку проїзду по відповідному маршруту, та 0 в протилежному випадку. Тоді $b_j^{out} = \sum_{i=1}^{m+2} b_{ij}$ – кількість виїздів з точки, географічні координати якої рівні $v[j,1], v[j,2]$, відповідно $b_i^{in} = \sum_{j=1}^{m+2} b_{ij}$ – кількість прибуттів до точки $v[i,1], v[i,2]$. Тоді задача оптимізації приймає вигляд:

$$\begin{cases} l = \sum_{i,j=1}^{m+2} a_{ij} b_{ij} \rightarrow \min \\ b_i^{in} - b_i^{out} = 0, i = \overline{2, m+1} \\ b_1^{in} - b_1^{out} = -1 \\ b_{m+2}^{in} - b_{m+2}^{out} = 1 \end{cases} \quad (24)$$

Як було показано автором [14], поле імовірності урбанізації зручно описати за допомогою апарату нечіткої логіки [15].

У загальному випадку імовірність P записуємо у вигляді

$$P = F(h, l), \quad (25)$$

де h, l – вхідні параметри, які визначаються згідно з (22) та (24) відповідно.

Як показали попередні дослідження [16, 17], найкращий рівень точності показав алгоритм нечіткого виведення Сугено [15]. За Сугено необхідно побудувати нечітку базу знань, кортежі якої матимуть вигляд:

$$(h = \tilde{a}_{1j} \Theta_j l = \tilde{a}_{2j}) \rightarrow y_j = b_{j0} + b_{j1}h + b_{j2}l, j = \overline{1, m}, \quad (26)$$

де \tilde{a}_{ij} — нечіткий терм, яким оцінюється змінні h, l в j -му правилі; Θ_j — логічна операція, що пов'язує фрагменти антецедента j -го правила; \rightarrow — нечітка імплікація; b_{j0}, b_{j1}, b_{j2} — деякі дійсні числа, що формують висновки j -го правила y_j ; m — кількість елементів нечіткої бази знань.

Ступіньналежностівхідноговектора $X^* = (h^*, l^*)$ до висновків нечіткої бази знань y_j визначають таким чином:

$$\mu_{y_j}(X^*) = \mu_j(h^*) \chi_j \mu_j(l^*), \quad j = \overline{1, m}, \quad (27)$$

де $\mu_j(x)$ — функція належності входу x нечіткому терму \tilde{a}_{ij} ; $\mu_{y_j}(X^*)$ — функція належності виходу y_j ; χ_j — t -норма.

У результаті нечіткого виведення для вхідного вектора X^* розраховуємо нечітку множину

$$\tilde{y} = \left(\frac{\mu_{d_1}(X^*)}{d_1}, \frac{\mu_{d_2}(X^*)}{d_2}, \dots, \frac{\mu_{d_m}(X^*)}{d_m} \right), \quad (28)$$

де множина нечітких термів $\{d_1, d_2, \dots, d_m\}$ — носій нечіткої множини \tilde{y} .

Вихідне значення y отримують у процесі дефазифікації \tilde{y} . За алгоритм дефазифікації в роботі використано зрівноважену суму [15]

$$y = \sum_{j=1}^m \mu_{y_j}(X^*) y_j, \quad (29)$$

де y — нечіткий висновок.

Для розрахунку форми поля імовірності урбанізації можна скористатися методом побудови карт рекреаційних потенціалів [16]. Для цього карта території T покривається прямокутником $\Pi = [a, b] \times [c, d]$. Очевидно, що прямокутник Π містить множину (територію) T ($T \subset \Pi$). Прямокутник Π розбивається сіткою $\Delta = \Delta_x \times \Delta_y$, де

$$\Delta_x = \bigcup_{k=0}^N \{x_k\}, \quad (30)$$

$$\Delta_y = \bigcup_{l=0}^M \{y_l\}, \quad (31)$$

$$x_k = x_0 + kh_x, \quad k = \overline{0, N}, \quad (32)$$

$$y_l = y_0 + lh_y, \quad l = \overline{0, M}, \quad (33)$$

$$h_x = \frac{b-a}{N}, \quad (34)$$

$$h_y = \frac{d-c}{M}, \quad (35)$$

де $(a, c), (b, d)$ — координати протилежних вершин прямокутника Π ; Δ_x, Δ_y — величина граней сітки; $\{x_k\}, \{y_l\}$ — координати вузлів сітки; N, M — кількість частин, на яку розбивається грань $[a, b]$ та $[c, d]$ прямокутника Π відповідно.

Для кожного вузла сітки визначаються значення вхідних параметрів. Отримані матриці є вхідними параметрами нечіткої функції поля імовірності урбанізації (1). Результатом розрахунку є матриця, яка визначає форму імовірності урбанізації T .

Отримане поле імовірності та обмеження враховуються на останніх кроках рекурсії побудови афінних фракталів, тобто в момент розрахунку координат будинку. Якщо географічні координати будинку знаходяться в околі обмежень, будинок не ініціалізується. В іншому випадку визначається імовірність побудови будинку $p(x, y)$ з заданими географічними координатами x, y з матриці T . Визначається з рівномірного розподілу випадкове число $r \in [0; 1)$. Якщо $r < p(x, y)$, будинок ініціалізується, в іншому випадку — ні. В результаті такого підходу будується стохастичний фрактал з розмірністю $1 < D < 2$. Положення будинків задається на останньому кроці рекурсії алгоритму у вигляді векторів $\vec{V}_h^i(z_{h1}^i, z_{h2}^i)$, $i = \overline{1, m}$, де m — кількість будинків.

В ході побудови фракталів на основі різних ініціаторів будинки можуть перекриватись. Тоді необхідно вилучити один з них у залежності від сфери впливу ініціатора. Перевагою підходу побудови форми населеного пункту на основі самоафінних фракталів є те, що з'являється можливість не тільки визначити координати будинків, але і їх орієнтацію. Тому в центрі міста, де спостерігається значне перекриття сфер впливу доріг, будинки можуть орієнтуватись вздовж різних ініціаторів. Це, на перший погляд, складає враження хаотичності в розбудові.

Нехай h_1 та h_2 відстані до першого та другого ініціатора від даних будинків, визначені на основі (20). Для вибору будинку, що вилучається, необхідно розрахувати нормовані відстані:

$$\bar{h}_i = \frac{h_i}{h_1 + h_2}, \quad i = 1, 2. \quad (37)$$

Визначається з рівномірного розподілу випадкове число $r \in [0; 1)$. Якщо $r < \bar{h}_1$ вилучається другий будинок. Тобто залишається будинок, орієнтований паралельно першому ініціатору. В іншому випадку вилучається перший будинок.

Для подальшого аналізу координати будинків зручно розташувати у матриці, аналогічно до поля імовірності. В загальному випадку дійсні та уявні частини вершин даних векторів не являються цілими числами. Зведення їх до цілочисельних індексів матриці призведе до втрати інформації про реальні напрямки будинків у просторі. Виходом з цієї ситуації є покриття досліджуваної території сіткою, якій відповідає комплексна матриця H . Комірці матриці заповнюються згідно з такими міркуваннями: якщо вершина вектора будинку належить комірці матриці, даній комірці присвоюються ко-

ординати другої вершини вектора будинку. Всім порожнім коміркам присвоюється нульове значення. Тобто в усіх ненульових комірках вказуються реальні координати будинків:

$$H(\text{int}(\text{Re}(z_{h1}^i)), \text{int}(\text{Im}(z_{h1}^i))) = z_{h2}^i, \quad i = 1, m \quad (36)$$

$$H(\text{int}(\text{Re}(z_{h2}^i)), \text{int}(\text{Im}(z_{h2}^i))) = z_{h1}^i,$$

де z_{h1}^i, z_{h2}^i – комплексні координати векторів будинків; $\text{Re}(\cdot), \text{Im}(\cdot)$ – дійсна та уявна частина комплексного аргументу; $\text{int}(\cdot)$ – функція округлення до цілого.

Якщо фрактал представлений у вигляді такої матриці, тоді фрактальну розмірність Хаусдорфа-Базікевича можна легко розрахувати згідно з [18]:

$$D(R) = 2 + \frac{\log(n_h/N)}{\log(R)}, \quad (37)$$

де R – радіус (лінійний розмір) досліджуваної частини матриці H ; N – кількість комірок, що містить досліджувана область; n_h – кількість непорожніх клітинок в досліджуваній області матриці H .

4. Алгоритм моделювання

Отже, запропоновано моделювати просторову форму туристичних поселень згідно з наступним алгоритмом (рис. 7).

Крок 1. Створення нечіткої бази знань.

Використовуючи ГІС системи, формується нечітка база знань (26). Для побудови навчальної множини згідно з (22) та (24) розраховуються значення h та l для випадкових географічних координат в околі реально існуючих туристичних поселень. В якості нечіткого висновку вказується рівень урбанізації території для вибраної точки. В результаті навчання формуються нечіткі терми та подукційні правила.

Крок 2. Формування масиву ініціаторів.

Використовуючи ГІС системи, формується масив векторів основних доріг населеного пункту. Визначається оптимальний розмір ініціатора у залежності від масштабу досліджуваної території. Кожен вектор дороги масштабується до розміру ініціатора за допомогою наступного афінного перетворення:

$$\vec{v}' = \frac{R}{|\vec{v}|} \cdot \left(\vec{v} - \frac{z_1 + z_2}{2} \right) + \frac{z_1 + z_2}{2}, \quad (38)$$

де z_1, z_2 – координати вектора дороги \vec{v} в комплексному вигляді; R – розмір ініціатора.

Крок 3. Розрахунок поля імовірності урбанізації та обмежень.

Використовуючи отриману нечітку базу знань, отриману на кроці 1, згідно з (29)-(35) розрахо-

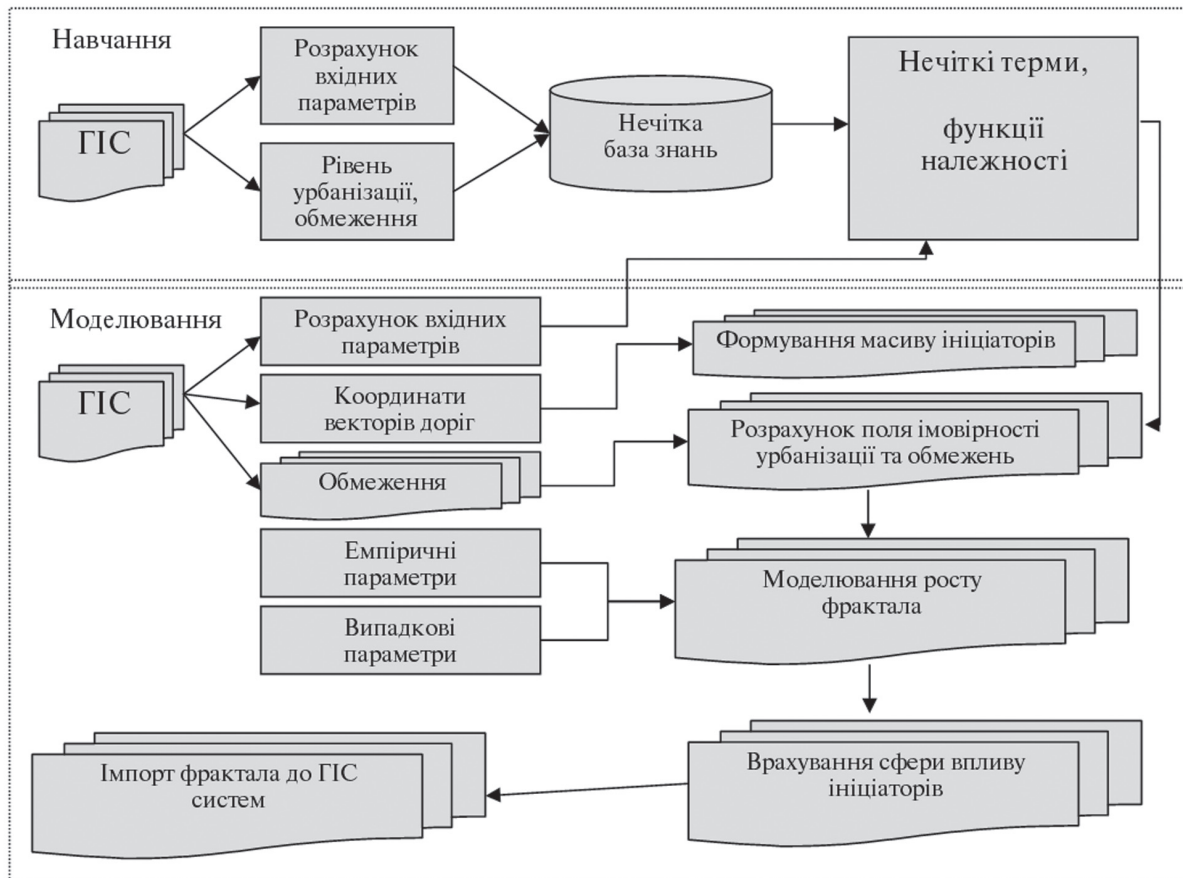


Рис. 7. Структурна схема алгоритму моделювання фрактальної структури туристичних поселень

вється матриця поля імовірності. За допомогою ГІС системи формуються матриці обмежень.

Крок 4. Моделювання фракталу.

За допомогою рекурсивних самоафінних фракталів (1)-(2), або (5)-(8) для кожного ініціатора будується фрактал з врахуванням поля імовірності та обмежень.

Крок 5. Врахування сфери впливу ініціаторів.

Згідно з вищенаведеним алгоритмом вилучаються близько розташовані будинки та будинки, що перетинаються, тобто ті, які потрапили у сферу впливу декількох ініціаторів.

Крок 6. Представлення результату.

Отриманий фрактал імпортується у ГІС системи у вигляді окремого про шарку.

5. Комп'ютерний експеримент

Для апробації моделі ми вибрали відоме курортне містечко українських Карпат — Ворохту (рис. 8, б). Як перше наближення під час розрахунку потенціального поля за вхідні параметри нечіткої моделі, основаної на алгоритмі нечіткого виводу Сугено, було вибрано координати гірськолижних витягів, віддалі до доріг і шлях дорогою до найближчого атрактора. Особливості рельєфу не враховували. Для формування продукційних правил нечіткої моделі використовувались ландшафтні дані урбанізованих територій Карпатського регіону, отримані за допомогою ГІС систем. Всього навчальна множина містила близько 1000 записів.

Розраховане навченою мережею нечітке імовірнісне поле представлено на рис. 8, а. З рисунку видно, що дане поле нагадує за формою урбанізовані території. В зоні максимальних забудов імовірність близька до 1. З віддаленням від центру міста імовірність спадає до 0. З рисунку видно, що градієнт спадання поля максимальний перпендикулярно до дороги. А вздовж доріг імовірність забудови спадає

повільніше, що підтверджують висновки роботи [3].

В розрахунках використовувався алгоритм побудови самоафінного фракталу (1)-(2). Рівень рекурсії складав 14. Довжина ініціатора складала $l=1000\text{м}$. Отже величина ініціатора 14-го рівня (тобто будинку) становить $l_{14} = S^{14} \cdot l_0 \approx 8\text{м}$. Час, затрачений на розрахунок становив $\sim 1\text{с}$ при частоті процесора Intel Core Duo 2000MHz. Отриманий після п'ятого кроку алгоритму фрактал наведений на рис. 8, в.

З порівняння отриманого фракталу та реальному населеному пункту можна прийти до висновку, що фрактал повторює основні особливості форми Ворохти. На під'їздах до містечка будинки розташовуються вздовж доріг, а в центрі спостерігається складна структура кварталів, зумовлена перекриттям зон впливу декількох ініціаторів.

Критерієм ступеня схожості форми стохастичного фракталу до модельованого об'єкту виступає співвідношення, запропоноване Мандельбротом [18]:

$$\rho = \frac{P^{1/D}}{S^{1/2}}, \quad (39)$$

де P — периметр багатогранника; D — фрактальна розмірність; S — площа багатогранника.

Якщо багатогранники подібні, то дане співвідношення для розрахованого фракталу (ρ_F) та населеного пункту (ρ_S) мають бути близькими.

Розрахована фрактальна розмірність отриманого стохастичного фракталу згідно з (37) становить $D \approx 1,384$. Для розрахунку фрактальної розмірності Ворохти територія покривалась сіткою, якій ставилась у відповідність матриця H' . Коміркам таблиці, яким відповідала урбанізована територія, присвоювалось значення 1, для неосвоєних територій — 0. Аналогічно розрахована фрактальна розмірність Ворохти становить $D \approx 1,273$. Близькість до одиниці фрактальної розмірності свідчить про

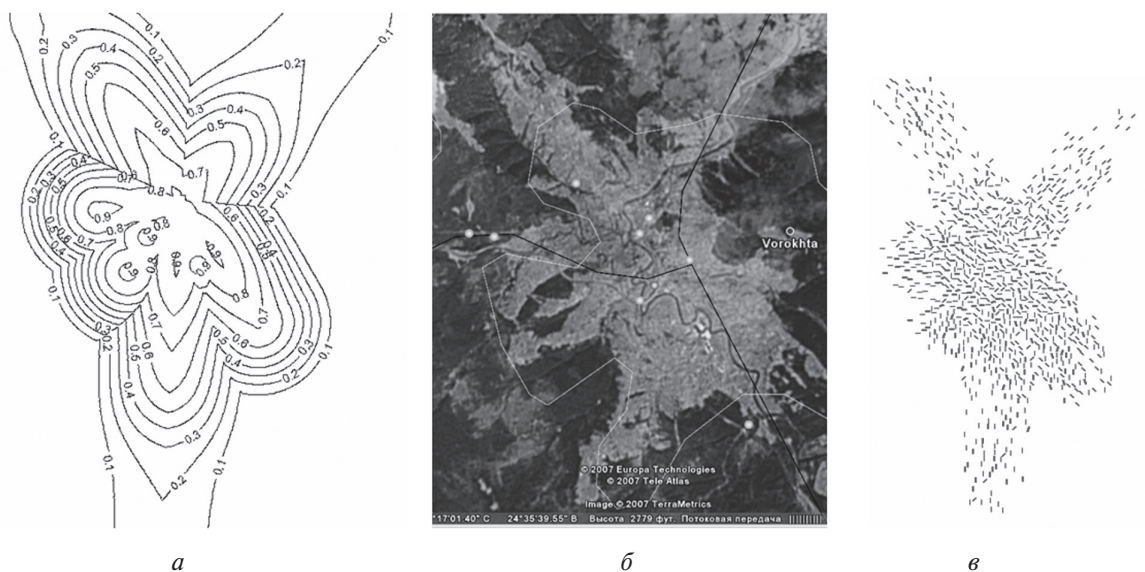


Рис. 8. Ворохта: а — нечітке поле імовірності урбанізації; б — світлина з космосу; в — розрахований фрактал

те, що структура міста ближче до лінійної, ніж до сферичної.

Периметри і площі побудованого фракталу та населеного пункту розраховувались шляхом підрахунку ненульових клітинок матриць H та H' . В результаті відхилення ρ_F від ρ_S становить 86%, що є підтвердженням адекватності та точності запропонованої моделі.

Висновки

У роботі наведено огляд можливих підходів до моделювання форм населених пунктів за допомогою самоафінних фракталів. Обґрунтовано математичні підходи до моделювання даних об'єктів. Представлено математичний вигляд афінних перетворень в комплексній формі. Наведено особливості розвитку та внутрішньої структури невеликих туристичних поселень.

Структурно представлено алгоритм моделювання фрактального росту поселень в нечіткому імовірнісному полі.

Обґрунтовано та наведено алгоритм розрахунку вхідних параметрів моделі. Представлено алгоритм побудови нечіткої бази знань та поля імовірності на основі нечіткого виводу Сугено. Запропоновано метод врахування імовірності забудови при рості самоафінного фракталу. Запропоновано метод врахування зони впливу декількох ініціаторів при моделюванні фрактального росту.

Розроблена методика моделювання фрактального росту населених пунктів представлена у вигляді покрокового алгоритму та структурної схеми.

Апробація алгоритму проводилась на туристичному курорті Українських Карпат – м. Ворохта. Отриманий фрактал за формою, розмірністю та ступенем схожості показав достатній рівень точності. Це підтверджує адекватність моделі.

До переваг такого підходу можна віднести можливість моделювання як форми, так і внутрішньої структури містечка за короткий проміжок часу. В той самий час, недоліком даного підходу є неможливість без додаткових наближень змоделювати динаміку зростання населеного пункту у часі.

Список літератури: 1. Alexander, C. A city is not a tree // *Architectural Forum*, 1965 – 122 (1) – с. 58-61 and (2) – с. 58-62. 2. Mandelbrot, B. B. Stochastic models for the earth's relief, the hape and fractal dimension of coastlines and the number-area rule for islands // *Proceedings of the National Academy of Sciences USA*, 1975 – 72 – с. 3825-3828. 3. Doxiadis, C. A. Ekistics: An Introduction to the Science of Human Settlements, Hutchinson, London, 1968. 4. Mandelbrot, B. B. Fractals - a geometry of nature // *New Scientist*, 1990 – 127 – с. 38-43. 5. Qingsheng Yanga, Xia Lia, Xun Shi Cellular automata for simulating land use changes based on support vector machines // *Computers & Geosciences*, 2008. – 34 – с. 592-602. 6. Teknomo K., Gerilla G.P., Hokao K. Cellular Urban Descriptors of Lowland Urban Model // *Proceedings of International Symposium of Lowland Technology*, Bangkok, September 2004, с. 297-302. 7. Elisabete S., Keith C. Complexity, emergence and cellular urban models: lessons learned from applying

SLEUTH to two Portuguese metropolitan areas // *European Planning Studies*. 2005, – 13,1 – с. 93-115. 8. Couclelis H. From cellular automata to urban models: new principles for model development and implementation // *Environment and Planning B: Planning and Design*. 1997, – 24(2) – с. 165-174. 9. Teknomo K., Gerilla G. P. and Hokao K. Stochastic cellular model for lowland urban development // *Lowland Technology Information Journal*. 2006. – 8, 1 – с. 1-10. 10. Reys, J. W, The Making of Urban America: A History of City Planning in the United States, Princeton University Press, Princeton, NJ. 1965. 11. M. Batty, P. Longley *Fractal cities*, Academic Press, London and San Diego, 1996, 394 p. 12. Кронове P.M. Фракталы и хаос в динамических системах / P.M. Кронове. – Москва: Техносфера, 2006, 488 с. 13. Keeble, L. Principles and Practice of Town and Country Planning, The Estates Gazette, London. 1959. 14. Вижлюк, Я. І. Методи побудови густини потенціального поля рекреаційної привабливості території [Текст] / Я.І. Вижлюк // Штучний Інтелект. – 2009. – № 2. – С. 151-160. 15. Заде Л.А. Понятие лингвистической переменной и его применение к принятию приближенных решений [Текст] / Л.А. Заде. – Москва: Мир, 1976. – 165 с. 16. Вижлюк, Я. І. Моделювання флуктуацій росту та сегментації соціально-економічних об'єктів у процесі фрактального росту в нечіткому потенціальному полі [Текст] / Я.І. Вижлюк // Інформаційні технології та комп'ютерна інженерія. – 2009. – № 1. – С. 23-32. 17. Вижлюк, Я. І. Методологія прогнозування соціально-економічних процесів методами фрактального росту кристалів у нечіткому потенціальному полі [Текст] / Я.І. Вижлюк // Вісник Тернопільського держ. техн. ун-ту. – 2008. – № 2. – С. 153-162. 18. Федер, Е. Фракталы [Текст] / Е. Федер; пер.с англ. – М.: Мир, 1991, 254 с.

Надійшла до редколегії 12.04.2010 р.

УДК 004.942, 004.891.2

Методы фрактальной геометрии и нечеткой логики, как основа методики моделирования пространственной организации туристических поселений / Я.І. Вижлюк // *Бионика интеллекта: науч.-техн. журнал*. – 2010. – №1 (72). – С. 109–117.

В статье предложена методика моделирования пространственной организации туристических поселений. Основой методики являются самоафинные стохастические фракталы и вероятностное поле на основе нечеткой логики. В работе проводится обзор основных работ зарубежных авторов и представлен весь необходимый для моделирования математический аппарат. Разработанная методика приводится в виде вербального и структурного алгоритма. Апробация модели при моделировании пространственной структуры г.Ворохта показала высокий уровень адекватности и точности.

Ил. 8. Библиогр.: 18 назв.

УДК 004.942, 004.891.2

Methods of fractal geometry and fuzzy logic as basis of modeling methodology of tourist settlements spatial arrangement. / Ya. Vyklyuk // *Bionics of Intelligence: Sci. Mag.* – 2010. – № 1 (72). – P. 109–117.

In this paper we represented methodology of tourist settlements spatial arrangement modeling. Basis of the methodology are selfeinsteinium stochastic fractals and stochastic field on the fuzzy logic basis. There are overviews of main papers of foreign authors and whole mathematical apparatus needed for modeling. Methodology was presented as verbal and structural algorithms. Model approbation on city Vorohtha showed high level of adequacy and accuracy.

Fig. 8. Ref.: 18 items.

УДК 681.3.016



ОРГАНИЗАЦИЯ ЗАПРОСОВ К РАСПРЕДЕЛЕННЫМ ДАННЫМ СРЕДСТВАМИ ЛОГИЧЕСКОГО ПРОГРАММИРОВАНИЯ

С.С. Тянянский¹, Ю.А. Мальков²¹ ХНУРЭ, г. Харьков, Украина, tanyansky_ss@yahoo.com;² ХНУРЭ, г. Харьков, Украина, malkov@smtp.ru

Рассматривается представление реляционных баз данных в виде множества логических фактов и правил. Описаны возможные варианты отличий в физической структуре отношений со схожим информационным наполнением. Рассмотрен метод создания совместного отображения данных с различной физической структурой, основанный на медиаторах и логическом программировании.

БАЗЫ ДАННЫХ, ФЕДЕРАЛИЗАЦИЯ, ДЕДУКТИВНЫЕ БАЗЫ ДАННЫХ, ИНТЕГРАЦИЯ БАЗ ДАННЫХ, ДЕЙТАЛОГ

Введение

В процессе развития информационных систем (ИС) крупных предприятий или организаций внедряется большое множество различных информационных подсистем. Как правило, эти подсистемы имеют свои собственные базы данных (БД) для накопления информации и последующего ее использования. В связи с информационной и логической связанностью бизнес-процессов предприятия, автоматизируемых данными подсистемами, часть информации неизбежно дублируется. Для исключения накопления одинаковой информации в различных подсистемах предприятия создают системы интеграции данных. Это позволяет использовать данные одной подсистемы другими. Для интеграции данных используют методы консолидации и федерализации. При консолидации данные переносятся в одну общую ИС, с общей структурой, управляемую реляционной системой управления базами данных (СУБД) промышленного уровня. После переноса данных все системы используют общую БД. Среди недостатков данной технологии необходимо отметить потерю данных при интеграции, а также значительные ресурсы памяти для их хранения. Другим подходом решения задачи интеграции данных является федерализация. Он позволяет не изменяя существующих ИС, извлекать данные из первичных источников и интегрировать их таким образом, чтобы они отвечали требованиям поступившего запроса.

На данный момент основным средством управления данными являются реляционные системы БД, использующие в качестве операционной спецификации алгебру множеств. Языки запросов, построенные на таких операциях, имеют ряд ограничений, таких как отсутствие рекурсивных функций и строгая привязка к структуре данных.

В настоящее время активно развиваются декларативные языковые средства в системах обработки данных.

Вопросами применения логического программирования в реляционных базах данных начали

рассматриваться еще в конце 70-х годов. Среди исследователей данной тематики можно выделить работы Дж. Малпаса [1], С. Чери и других [2], Стерлинга и Шапиро [3], Девяткова В.В. [4], в которых описана методика применения логических языков, и в частности Дейталога, в качестве языка определения и манипулирования данными.

Вопрос о применении средств логического программирования для реализации системы интеграции данных в литературе ранее не рассматривался. В связи с этим целью настоящей статьи является исследование и разработка метода приведения БД с неоднородной структурой к единому виду, используя интенциональные свойства таблиц. Также предложен подход к организации поиска данных средствами логического программирования.

Ниже рассматривается методика совместного использования данных при помощи средств логического языка Дейталога, поддерживаемого дедуктивными базами данных (ДБД).

1. Применение логических языков для манипулирования объектами БД

Рассмотрим механизм ДБД для реализации системы интеграции данных, основанной на федерализации. Это позволит оперировать объектами базы данных при помощи логических выражений, а сами объекты базы данных описывать логическими фактами и правилами. В качестве языка используем язык логического программирования Дейталога, разработанный специально для БД [2, 4].

Отношение БД будем именовать предикатом и представлять в виде набора базовых дизъюнктов (фактов) вида (1):

$$t(a_1, a_2, \dots, a_n), \quad (1)$$

где, t – имя предиката, а a_1, a_2, \dots, a_n – аргументы предиката, определяющие его аридность.

Факты являются входными данными Дейталога-программы и все их множество представляют экстенциональной базой данных (ЭБД).

Кроме ЭБД существует интенциональная база данных (ИБД) – результат выполнения Дейталога-

программы. ИБД строится из ЭБД с помощью правил, определяющих ее содержимое.

Правило – это конструкция вида

$$q :- p_1, \dots, p_n, \quad (2)$$

разделенная символом условной дизъюнкции “:-”. Слева от символа “:-” находится голова правила, а справа – тело. Приведенное правило означает, что цель q будет истинной, если будут истинны факты p_1, \dots, p_n , стоящие в теле правила.

В Дейталоге при помощи правил осуществляют запросы данных из множества фактов, а также создают новые логические конструкции, с которыми в дальнейшем можно работать, как с фактами.

При построении сложных правил часто используются переменные – строки буквенно-цифровых символов конечной длины, начинающиеся с прописной буквы. Например X , Query_2 и так далее. Они необходимы для подстановки атрибутов предиката из тела правила в его голову, а также задания условий в теле правила. Необходимо отметить, что переменные, которые появляются в голове правила, обязательно должны упоминаться в теле правила.

2. Система федерализации данных, основанная на медиаторах

Медиаторы – программные компоненты, обеспечивающие поддержку виртуальных баз данных, которые транслирует каждый запрос пользователя в один или несколько запросов, адресованных различным СУБД [5]. По результатам обработки частных запросов медиатор синтезирует ответ на исходный запрос.

Рассмотрим общую схему построения системы интеграции БД с применением медиаторов (рис. 1).

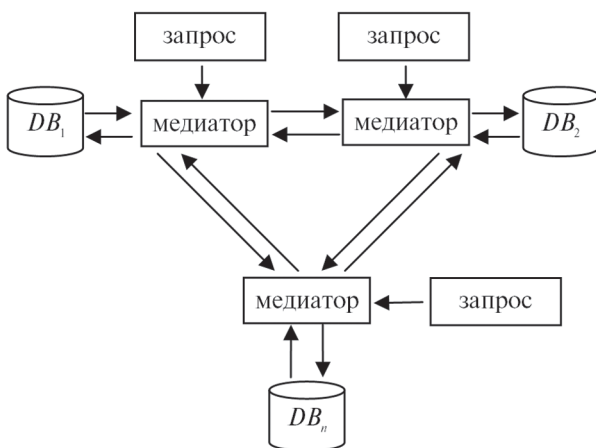


Рис. 1. Схема системы интеграции данных на основе медиаторов

Представим информационную систему предприятия в виде множества баз данных информационных подсистем $DB = \{DB_1, \dots, DB_n\}$. Некоторые отношения в этих базах имеют схожее информационное наполнение, но между ними существуют

различия в физической структуре, обусловленные различными вариантами реализации схемы при проектировании и различной степенью нормализации.

Рассмотрим схему построения медиатора для произвольной БД с применением средств логического программирования (рис. 2).

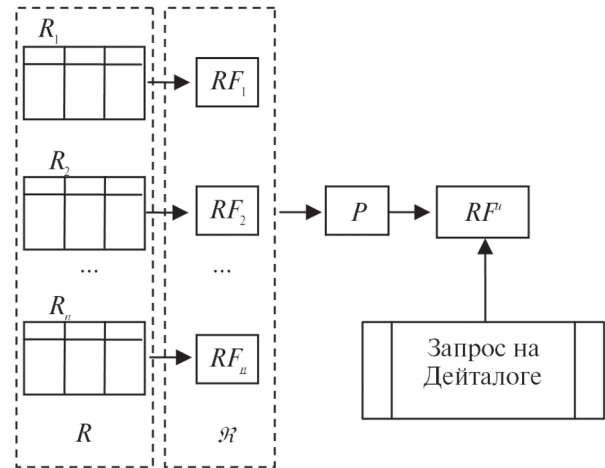


Рис. 2. Схема построения медиатора с применением логического программирования

Представим каждое отношение базы данных $R = \{R_1, R_2, \dots, R_n\}$ в виде множества $\mathfrak{R} = \{RF_1, RF_2, \dots, RF_n\}$, где RF_i – множество фактов вида (1), и определим логические правила $P = \{P_1, P_2, \dots, P_m\}$, где P_i – выражения вида (2), которые позволяют выделить интенсионал $RF^m = \{rfu_1, rfu_2, \dots, rfu_k\}$ для отношений R , где rfu_i – выражение вида (1). Представленная схема построения медиатора позволяет формировать запросы к БД различной структуры с использованием языковых средств Дейталога.

3. Построение интенсионала интегрируемых БД

Рассмотрим варианты приведения различных отношений R_i к общему интенсионалу RF^m на примере интеграции отношений баз данных учета успеваемости студентов. Первым и возможно самым распространенным вариантом различий между отношениями являются отличия в физической структуре схемы БД, то есть различная степень нормализации.

Рассмотрим процедуру создания интенсионала RF^m для БД DB_1 и DB_2 , имеющих различные схемы. Пусть R_1 – некоторое отношение из DB_1 (рис. 3), а R_2 и R_3 – отношения из DB_2 (рис. 4, 5), связанные по полю id.

фамилия	предмет	дата	группа	оценка
Иванов	математика	2008-11-01	ИС-01	4
Петров	английский	2008-12-04	ЗИ-02	5
Сидоров	математика	2008-12-21	ИС-03	3
Иванов	английский	2008-12-28	ИС-01	3
Сидоров	английский	2008-12-23	ПМ-01	2

Рис. 3. Отношение R_1

фамилия	курс	группа	id
Пушкин	4	пм-2	2
Лермонтов	2	иб-3	3
Есенин	1	пм-3	4

Рис. 4. Отношения R_2

id	предмет	оценка
2	математика	2
3	английский	5
4	английский	3
3	украинский	4

Рис. 5. Отношения R_3

Представим отношения R_1, R_2, R_3 в виде множеств логических фактов $r1, r2$ и $r3$ вида (1). Для этого каждый кортеж исходного отношения запишем в виде базового дизъюнкта, аргументами которого будут выступать значения атрибутов кортежа.

- $r1$ (“Иванов”, “математика”, 2008-11-01, “ИС-01”, 4).
-
- $r1$ (“Сидоров”, “английский”, 2008-12-23, “ПМ-01”, 2).
- $r2$ (“Пушкин”, 4, “пм-2”, 2).
- ...
- $r2$ (“Есенин”, 1, “пм-3”, 4).
- $r3$ (2, “математика”, 2).
- ...
- $r3$ (3, “украинский”, 4).

Определим отношение R_1 как интенционал RF^u для отношений R_1, R_2, R_3 . Чтобы привести отношения R_2 и R_3 к виду RF^u , определим следующее правило:

- $rfu(X, Y, G, Z, K):-r2(X, _, G, I),$
- $r3(I, Y, K), Z=\text{“нет_данных”}.$

В данном правиле переменная I связывает отношения R_2 и R_3 по полю id. Переменные X, Y, G, K переносят в структуру результирующего множества (цели) RF^u атрибуты “фамилия”, “предмет”, “группа”, “оценка” отношений R_2 и R_3 ; при этом переменная Z добавляет атрибут “дата” для установления эквивалентности схем $RF^u \equiv R_1 \equiv R_2 \cup R_3$.

Кроме проблемы различий в физической структуре таблиц существуют также различия в уровнях абстракции при проектировании БД. Так, типы элементов данных, имеющих несколько отличающихся подтипов, можно представить как отдельными таблицами для каждого подтипа, так и одной таблицей для всех подтипов, введя при этом столбец-селектор, определяющий значение подтипа. Для примера рассмотрим отношения R_4 (рис. 6) и R_5 (рис. 7) с данными об успеваемости двух групп “ПМ-04” и “ИС-01”. Для данной структуры характерным признаком является вынесенный элемент данных “группа” в заголовок отношения, что не соответствует структурам других отношений и усложняет процесс обработки данных.

ПМ-04			
фамилия	дата	предмет	оценка
Шевченко	2008-12-01	физика	3
Петренко	2008-12-21	биология	5
Ищенко	2008-12-26	химия	3

Рис. 6. Отношение R_4

ИС-01			
фамилия	дата	предмет	оценка
Тищенко	2008-11-07	алгебра	4
Зозуля	2008-12-01	геометрия	2
Яценко	2008-12-06	физика	5

Рис. 7. Отношение R_5

Чтобы привести отношения R_4 и R_5 к интенционалу RF^u , запишем их в виде множества логических фактов вида (1) и определим следующие правила:

- $rfu(X, Y, Z, G, K):-r4(X, Z, Y, K), G=\text{“пм-04”},$
- $rfu(X, Y, Z, G, K):-r5(X, Z, Y, K), G=\text{“ис-01”}.$ (3)

В правилах (3) переменные X, Y, Z, K переносят атрибуты “фамилия”, “дата”, “предмет”, “оценка” отношений R_4 и R_5 в результирующий набор данных RF^u , а с помощью переменной G вводится отсутствующий в отношениях R_4 и R_5 атрибут “группа”.

Рассмотрим вариант приведения к интенционалу RF^u , кросс-таблицы R_6 (рис. 8).

группа	фамилия	дата	математика	физика	химия	биология
иб-03	Петренко	2008-10-11	3	5	3	4
иб-03	Евтушенко	2008-11-03	4	5	5	2
иб-02	Симоненко	2008-11-04	4	4	3	4

Рис. 8. Отношение R_6

Представим отношение R_6 как множество логических фактов и определим для него правила приведения R_6 к виду RF^u .

- $rfu(X, Y, Z, G, K):-r6(G, X, Z, K, _, _, _), Y=\text{“математика”},$
- $rfu(X, Y, Z, G, K):-r6(G, X, Z, _, K, _, _), Y=\text{“физика”},$
- $rfu(X, Y, Z, G, K):-r6(G, X, Z, _, _, K, _), Y=\text{“химия”},$
- $rfu(X, Y, Z, G, K):-r6(G, X, Z, _, _, _, K), Y=\text{“биология”}.$

В приведенных правилах переменные G, X, Z переносят атрибуты “группа”, “фамилия”, “дата” отношения R_6 в RF^u . С помощью переменной K выбирается значение “оценки”, а Y передает название столбца, соответствующего этой оценке. Символ подчеркивания на месте атрибута обозначает, что этот атрибут не будет присутствовать в конечном множестве.

4. Реализация запросов средствами Дейталог

Для применения аппарата логического программирования к задаче интеграции данных при помощи медиаторов представим каждую БД ис-

ходного множества ИС в виде множества логических фактов вида (1). Полученные множества фактов образуют экстенционал этих БД. Для реализации запросов ко всему множеству БД с различной структурой сформируем интенционал этих БД с помощью логических правил вида (2). Данные правила приведут структуры исходных экстенционалов к единому интенционалу RF^u . Это позволит формировать запросы ко всему распределенному множеству БД.

Запрос данных из множества логических фактов в Дейталоге формируют при помощи логических правил вида (2). Рассмотрим процедуру составления запроса (цели) к множеству логических фактов и правил. В качестве источника данных будем использовать интенционал отношений RF^u , который имеет следующую структуру:

$$\begin{aligned} &rfu(x_{11}, x_{12}, x_{13}, \dots, x_{1k}), \\ &rfu(x_{21}, x_{22}, x_{23}, \dots, x_{2k}), \\ &\dots \\ &rfu(x_{m1}, x_{m2}, x_{m3}, \dots, x_{mk}). \end{aligned} \quad (4)$$

Используя интенционал вида (4), можно формировать запросы вида (5), которые будут использовать множество распределенных БД как единый консолидированный источник данных.

$$q(X1, X2, X3, \dots, XK) :- rfu(X1, X2, X3, \dots, XK), X3=1123. \quad (5)$$

Здесь rfu – исходное множество данных, $X1, X2, X3, \dots, XK$ – набор переменных, переносящих атрибуты rfu в результирующий набор q , а $X3=1123$ условие выборки.

Выводы

В силу ряда причин между схожими по информационному наполнению БД часто существуют отличия в их физической структуре, обусловленные различной степенью нормализации и отличными вариантами реализации схемы. Это делает невозможным использовать множество БД как источник информации для запросов, без внесения изменений в исходные отношения.

Описанная технология интеграции информационных систем на основе медиаторов, реализованных средствами логического программирования, позволяет решить задачу интеграции неоднородных БД за счет приведения исходных отношений к общему интенционалу с помощью логических фактов и правил.

Практическая значимость полученных результатов заключается в возможности создания уни-

версального средства формулирования запросов к БД с различной структурой. В рассматриваемом материале не учитывается синтаксическая несогласованность атрибутов и предполагается, что эта задача решена на этапе формирования метаданных, которые используются при согласовании элементов правил в медиаторах. Данный материал выходит за рамки исследований, проведенных в работе и представленных в статье.

Полученные результаты дают основания для дальнейшего изучения свойств логических выражений языка Дейталога для доказательства корректности выполняемых операций над данными. При этом необходимо рассматривать совокупность запросов в виде набора правил, выполняемых как одна транзакция.

Список литературы: 1. *Маллас, Дж.* Реляционный язык Пролог и его применение [Текст] : пер. с англ. – М. : Наука, 1990. – 464 с. 2. *Чери, С.* Логическое программирование и базы данных [Текст] : пер. с англ. / С. Чери, Г. Готлоб, Л. Танка. – М. : Мир, 1992. – 352 с. 3. *Стерлинг, Л.* Искусство программирования на языке Пролог [Текст] : пер. с англ. / Л. Стерлинг, Э. Шапиро. – М. : Мир, 1990. – 333 с. 4. *Девятков, В.В.* Системы искусственного интеллекта [Текст] / В.В. Девятков. – М. : Издательство МГТУ им. Н.Э. Баумана, 2001. – 352 с. 5. *Корнеев, В.В.* Базы данных: интеллектуальная обработка информации [Текст] / В.В. Корнеев, А.Ф. Гореев, С.В. Васютин, В.В. Райх. – М. : Нолидж, 2001. – 496 с.

Поступила в редколлегию 14.04.2010 г.

УДК 681.3.016

Організація запитів до розподілених даних засобами логічного програмування / С.С. Танянський, Ю.А. Мальков // Біоніка інтелекту: наук.-техн. журнал. – 2010. – № 1 (72). – С. 118–121.

У статті розглянуто питання організації доступу до розподілених даних засобами логічного програмування. Даний підхід дозволяє розв'язувати задачі обробки гетерогенних даних. Запропоновано технологію об'єднання інтенціональних та екстенціональних баз даних в одну структуру, для якої формуються запити споживача.

Л. 8. Бібліогр.: 5 найм.

UDK 681.3.016

Building queries for distributed data via methods of logical programming. / S.S. Tanyansky, Y.A. Malkov // Bionics of Intelligence: Sci. Mag. – 2010. – № 1 (72). – P. 118–121.

The aspects of accessing distributed data via methods of logical programming have been viewed in the article. This method allows to complete the tasks of heterogeneous data processing. The technology of uniting the intensional and extentional databases in one structure, for which the client's queries are formed, has been offered.

Fig. 8. Ref.: 5 items.

УДК 681.324.01



ЗАДАЧИ И КОНЦЕПЦИИ МЕТОДОВ ПРИНЯТИЯ МНОГОКРИТЕРИАЛЬНЫХ РЕШЕНИЙ В ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМАХ

Г.И. Стопченко¹, И.А. Макрушан², С. В. Билан³

¹ г. Харьков, Украина, tutu@vk.kh.ua

² ХНУРЭ, г. Харьков, Украина, mivenera@mail.ru

³ ХНУРЭ, г. Харьков, Украина, belka-tyan@mail.ru

Произведен анализ существующих подходов к структуризации методов принятия многокритериальных решений. Выделен ряд классификационных признаков, позволяющих расширить существующую модель принятия решений. Предлагается иерархическая структура классификации существующих и разрабатываемых методов принятия многокритериальных решений, а также система концепций поиска решений.

ИНТЕЛЛЕКТУАЛЬНАЯ СИСТЕМА, МНОГОКРИТЕРИАЛЬНАЯ ЗАДАЧА, ПРИНЯТИЕ РЕШЕНИЙ, КОНЦЕПЦИЯ, МЕТОД

Введение

Задача принятия решений — одна из самых распространенных в любой предметной деятельности человека. Она часто формулируется как оптимизационная задача в терминах определенного математического аппарата и сводится в общем случае к выбору одной или нескольких альтернатив из некоторого множества. Большинство прикладных задач являются многоцелевыми, и функция принятия решений непосредственно связана с целеустремленной деятельностью человека.

В задачах многокритериального принятия решений для устранения неопределенности, возникающей из-за наличия нескольких критериев, используются предпочтения лица, принимающего решения (ЛПР), основанные на его знаниях, опыте и интуиции. Эта субъективная информация является основой объединения основных параметров задачи в единую модель, позволяющую оценивать варианты решений.

К настоящему времени разработано значительное количество моделей процедур многокритериального принятия решений. Несмотря на специфику этих средств, в каждой из них решения предъявляются для анализа процедур ЛПР, и после оценки информация в виде некоторых новых значений параметров используется для генераций, предположительно улучшаемых решений. Можно выделить некоторый набор операций, выполняемых ЛПР в процессе выявления его предпочтений: ЛПР определяет направление улучшения и величину шага в критериальном пространстве; задает весовые коэффициенты важности критериев; задает желаемые и пороговые уровни значений критериев; выделяет критерии, оценки по которым могут быть ухудшены или должны быть улучшены, проводит попарное сравнение решений в пространстве значений критериев и так далее.

В существующих обзорах по анализу и классификации моделей, методов и алгоритмов принятия

решений основным признаком выделения и классификации математических средств является тип дополнительной информации ЛПР. Разнообразие методов обусловлено множеством форм предоставления дополнительной информации от ЛПР и требует рассмотрения условий и особенностей принятия решений ЛПР, эффективности получаемых решений для выявления основных подходов [1-5].

1. Постановка задачи

Несмотря на значительное количество обзоров, методов и процедур принятия решений можно сделать следующие выводы:

- наблюдается некоторая произвольность в выборе классификационных признаков и отсутствие системобразующего концептуального элемента метода поиска решения;
- разнообразие форм дополнительной информации от ЛПР приводит к произвольной классификации;
- отсутствует четкая постановка автономности классов метода;

В этих условиях представляется целесообразным:

- введение дополнительных признаков классификации и автономности методов и моделей;
- на основе введения дополнительных признаков обеспечить лучшую структуризацию математических средств;
- обосновать автономность и типизацию разнообразных классов математических средств принятия решений.

2. Модель компонентов многокритериальных задач

Принятия решений — это особый вид целенаправленной деятельности, заключающийся в выборе наиболее эффективной из имеющихся альтернатив. Наиболее предпочтительной в методологии принятия решений является следующая модель:

$$M_1 = \langle t, x, D, F, H, P, r, S \rangle, \quad (1)$$

где t – вид требуемого решения; x – вектор переменных решений; D – область допустимых решений; F – вектор критериев оценки вариантов; H – способы измерения предпочтений (обычно шкалы); P – система предпочтений ЛПР (совокупность неформализованных представлений эффективности альтернатив); r – решающее правило (принцип оптимальности – формализованное выражение предпочтений ЛПР, представляющее собой принцип сравнения векторных оценок): $S = X \rightarrow Z$ – отображение решений в критериальные оценки.

Модель (1) является достаточно общей для широкого класса постановок многокритериальных задач (МКЗ). Представляется целесообразным шагом дальнейшая структуризация модели (1) путем выделения ряда дополнительных классификационных признаков (компонентов) для типизации концептуальных свойств:

- концепция построения модели (оптимизации, удовлетворения, приближения и достижения);
- тип постановки задач;
- наличие у ЛПР области желаемых значений (ОЖЗ) критериев, что предполагает возможность ЛПР выразить требования к получаемому решению;
- свойство области допустимых решений (ОДР) D : неизменность области D или возможность изменения области $D + \Delta D$;
- принадлежность получаемого наилучшего (желаемого) решения ОДР ($x^* \in D$);
- оператор поиска решения: максимизация (max), минимизация (min), достижения (equal);
- принцип оптимальности концептуальных моделей, который является формализованным выражением предпочтений ЛПР и представляют собой правило сравнения векторных оценок;
- условия поиска решения;
- основные (управляемые) параметры модели.

3. Концепции методов принятия решений

Различные сочетания элементов задачи приводят к многообразным задачам выбора, которые исследованы и изучены в разной степени. Многокритериальные задачи не имеют однозначного общего решения, поэтому существует много разных способов придать многокритериальной задаче частный вид, обладающий единственным решением. Естественно, что разные способы этих решений в общем случае оказываются различными, поэтому существенным аспектом в решении многокритериальной задачи является обоснование именно данного вида ее постановки. В зависимости от возможностей, предпочтений и наличия уровня притязаний (желательных целевых значений критериев), а также структуры ОДР D и типов постановок решаемых

задач, сформулируем типы концептуальных задач поиска решений [6,7]. Концепция оптимизации заключается в поиске априори неизвестного, наилучшего решения на основе существования функции полезности, которой руководствуется ЛПР при выборе наиболее предпочтительного решения, и у него нет представления о его желательных свойствах [8,9].

Задача формулируется следующим образом: найти вектор X , принадлежащий ОДР D и оптимизирующую совокупность целевых функций $\{f_i(x)\}$ при наиболее предпочтительном соотношении между их значениями в точке решения. Это требование означает нахождение в ОДР D эффективного решения X^* , соответствующего экстремуму априори неизвестной функции полезности $V(Z)$ ЛПР:

$$X^* \{ \text{arg opt } V(Z(X)) \}. \quad (2)$$

Методы решения (2) на основе концепции оптимизации имеют следующие особенности: у ЛПР отсутствует ОДР критериев; оператор поиска решений (opt) в зависимости от физического смысла критериев может быть max или min; ОДР постоянна; оптимальное решение принадлежит ОДР ($X^* \in D$); управляемыми параметрами, как правило, являются коэффициенты важности критериев $\Lambda = (\lambda_1, \lambda_2, \dots, \lambda_n)$. Принцип оптимальности ϕ_1 представляется следующим образом: $\phi_1: \{ \text{opt } (F_1(x, \Lambda)) \} \rightarrow X^*$. Самый известный и простой подход взвешенной свертки – использование аддитивной функции полезности в виде $V(F_1(x, \Lambda))$. Вид функции задается априори (часто линейный), а параметрам $\Lambda = \{\lambda_i\}$ придаются некоторые начальные значения Λ^0 , и решается задача $\text{opt } F\{f_i(x), \lambda_i\}; x \in D$. Полученные значения критериев рассматриваются ЛПР, и в случае неудовлетворенности решением параметры меняются в соответствии с некоторыми принципами.

На этой концепции оптимизации основываются следующие методы: Дайера – Джонсона, Сайвина; процедуры прогрессивной ориентации, 3–В, S MOLP, IMOLP и другие.

Концепция удовлетворения является основой моделей и методов многокритериального принятия решений, в которых осуществляется поиск альтернатив, критериальные оценки которых устраивают ЛПР. Эти требования к альтернативам могут априори быть у ЛПР или вырабатываться по мере ознакомления с ОДР.

Задача формализуется следующим образом: найти X , принадлежащий ОДР D и удовлетворяющий множеству критериальных оценок $L = \{l_i\}$:

$$\begin{aligned} \text{opt } F(x) &= \{f_i(x)\}; \\ X \in D &= \{A_x \geq B\}; \\ \{f_i(x) &\geq l_i\}. \end{aligned}$$

Методы, основанные на концепции удовлетворения, имеют следующие особенности: у ЛПР имеются требования к критериальным оценкам ОЖЗ; ОДР постоянна, область поиска решения постоянна (*Egual* - равенство), что отражает принадлежность лучшего решения ОДР ($X^* \in D$); управляемыми параметрами являются критериальные ограничения $L = \{l_i\}$ и уступки Δl . Принцип оптимальности φ_2 представим следующим образом:

$$\varphi_2 : \{F(f_i(x) \text{ equal } L = \{l_i\}) \rightarrow X^* .$$

Примером метода поиска удовлетворительного решения является метод уступок, в котором ЛПР ранжирует критерии по важности, затем решается задача оптимизации первого критерия при $X \in D$, затем второго и так далее, то есть определяется лексикографически оптимальное решение. Если полученное решение не устраивает ЛПР, ему необходимо указать, на какую величину Δ_1 он согласен снизить значение первого критерия для улучшения значения второго, и решается новая задача:

$$\begin{aligned} f_2(x) &\rightarrow \max; \\ f_1(x) &\geq Z_1^* - \Delta_1; \\ X &\in D. \end{aligned}$$

Если решение X_2 не обеспечивает приемлемого значения f_3 , ЛПР определяет уступку по второму критерию Δ_2 , и решается задача:

$$\begin{aligned} f_3(x) &\rightarrow \max; \\ f_1(x) &\geq Z_1^* - \Delta_2; \\ f_2(x) &\geq Z_2^* - \Delta_2; \\ X &\in D. \end{aligned}$$

Аналогично формируется задача по другим критериям.

На концепции удовлетворения основаны методы и процедуры: ISWT (Чанконга-Хеймса), Беленсона-Капура, внешнего ветвления, метод ϵ -ограничений, STEM, IMGP, SEMOPS и другие.

Концепция приближения заключается в целенаправленном формировании допустимых решений, когда цели и требования формируются ЛПР в значениях критериального пространства [10,11].

Задача формируется следующим образом: найти вектор X , принадлежащий ОДР D и обеспечивающий для целевых функций наиболее близкое приближение к множеству одновременно недопустимых значений целей $\hat{Z} = \{\hat{z}_i\}$:

$$X = \text{agr} \min d(F(x), \hat{Z}),$$

где $\hat{Z} = (Z_1, \dots, Z_n)$ – значения критериальных целей; $d(\cdot)$ – расстояние, определяемое на основании исходной метрики.

Общую модель представим в следующем виде:

$$d_p = \left[\sum_{i=1}^n \lambda_i |f_i(x) - \hat{z}_i|^p \right],$$

где p – норма измерения. Методы, основанные на этой концепции, имеют следующие особенности: у ЛПР имеются требования к критериальным оценкам ОЖЗ; ОДР постоянна; полученное решение принадлежит ОДР $D(X^* \in D)$, оператор поиска \min ; ОЖЗ и ОДР не пересекаются; основные управляемые параметры \hat{Z}, p, Λ . Принцип оптимальности φ_3 представляется в следующем виде:

$$\varphi_3 : \{\hat{Z}, \min d(F(x), \Lambda, \hat{Z})\} \rightarrow X^* .$$

При формализации целевого программирования изменяется модель задачи: к исходным условиям добавляются целевые ограничения, отражающие уровни ОЖЗ; вводятся переменные от желаемых значений исходных критериев t ; критерий в модели строится как функция новых переменных [12].

В модели цель выражается минимизацией переменных-отклонений t_i , которые отражают недопущение \hat{t}_i или превышение t_i^t целевого уровня \hat{Z}_i , в качестве целевой функции может использоваться линейная свертка:

$$\min \sum_{i=1}^n (\bar{\lambda}_i \bar{t}_i + \lambda_i t_i).$$

На концепции приближения основываются методы: достижения цели, изменяемого идеала; контрольной точки, SEMOPS, STEM, SIGMOP, компромиссного программирования и так далее.

Концепция достижения заключается в целенаправленном формировании допустимых решений при варьируемой (расширяемой) структуре ограничений ОДР [13].

Задача формируется следующим образом: найти X , удовлетворяющий множеству критериальным оценкам \hat{Z} директивной области Y за счет расширения ОДР D .

Модель задачи системной оптимизации:

$$(A^0 + \Delta A)\hat{x} \leq b^0 + \Delta b, \Delta A \in A, \Delta b \in b,$$

где ΔA – вариации коэффициентов матрицы A ; Δb – вариации правых частей ограничений; \hat{X} – целевая установка (\hat{X} не удовлетворяет при $\Delta A = 0, \Delta b = 0$).

Методы и процедуры, основывающиеся на этой концепции, имеют следующие особенности: у ЛПР имеется директивная ОЖЗ, выполнение и достижение которой обязательно, и ОДР, способная расширяться. Полученное решение не принадлежит первоначальной ОДР ($X^* \notin D$), а принадлежит расширенной $D + \Delta D$.

Оператор поиска решения – *Egual* (достижение), то есть пересечение директивной области и расширенной ОДР не пусто.

Управляемые параметры представляют собой изменяемые коэффициенты ΔA и правые части

Δb ограничений ОДР. Принцип оптимальности φ_4 можно представить следующим образом:

$$\varphi_4 : \{F(x, \Delta D) \text{ equal } \hat{Z}^{dup}\} \rightarrow X^* .$$

В общем виде структуризацию процесса поиска решений можно представить в виде следующей иерархической структуры (рис.1).

Объединенная модель принятия решений		
Концепции		
Приближения	Оптимизации	
Достижения	Удовлетворения	
Методы и процедуры		
Управляемые параметры		
Типы параметров		
Точечные	Нечетные	Интервальные

Рис. 1. Иерархическая структура классификации существующих и разрабатываемых методов принятия многокритериальных решений

Выводы

Анализ типов постановок задач принятия МК решений позволил определить ряд специфических свойств, позволяющих определить концептуальность математических средств ПР.

Выделение концептуальности методов способствует формализации постановок задач, выбору соответствующего метода решения и формированию процедур поиска решения, в том числе комбинированного типа.

С учетом концептуальности математических средств можно выделить следующую иерархическую структуру классификации: обобщенная модель ПР – концепции – методы и процедуры – управляемые параметры.

Выделение классов методов на основе концептуальности способствуют формированию для каждого класса возможности математического обоснования, зависимости эффективности получаемых решений от параметров модели.

Предлагаемый подход использовался при управлении бизнес-процессами, сетевыми информационными технологиями и при проектировании информационных систем.

Список литературы: 1. *White D.J.* A bibliography on the application of mathematical programming multiple-objective methods // *J. Oper. Res. Soc.* – 1990 – №41. – pp. 669-691. 2. *Ларичев, О.Н.* Аналитический обзор процедур решения многокритериальных задач математического программирования [Текст] / О.Н. Ларичев, А.Д. Никифоров // *Экономика и математические методы.* – 1986. Т. XXII, вып. 3. – С. 508-523. 3. *Штойер, Р.Е.* Процедуры интерактивной многокритериальной оптимизации: Чебышевская, желаемого вектора критериев и комбинированная [Текст] / Р.Е. Штойер // *Автоматика.* – 1993. №5. – С. 14-25. 4. *Chankong V. and Haimes Y.Y.* Multiobjective Decision Making: Theory and Methodology. - New York: North-Holland, 1983. – 406 p. 5. *Эддоус, М.* Методы принятия решений [Текст] / М. Эддоус, Р. Стэнсфильд. – М.: Аудит, ЮНИТИ, 1997. – 590 с. 6. *Левыкин, В.М.* Комплексная многокритериальная модель решения задач управления сетевыми информационными технологиями [Текст] / В.М. Левыкин, Г.И. Стопченко, И.А. Макрушан // *Нові технології (Науковий вісник КУЕІТУ).* – 2008. – №1 (19). – С. 142-148. 7. *Стопченко, Г.И.,* Интеллектуальные СППР в управлении сетевой информационной технологией [Текст] / Г.И. Стопченко, И.А. Макрушан // *Прикладная радиоэлектроника.* – 2008. – Т.7, №1, с. 43-53. 8. *Ларичев, О.И.* Теория и методы принятия решений [Текст] / О.И. Ларичев. – М.: Логос, 2002. – 392 с. 9. *Овезгельдыев, А.О.* Синтез и идентификация моделей многофакторного оценивания и оптимизации [Текст] / А.О. Овезгельдыев, Э.Г. Петров, К.Э. Петров. – К.: Наукова думка, 2002. – 164 с. 10. *R.E. Steuer, P.Na.* Multiple criteria decision making combined with finance: A categorized bibliographic study // *European Journal of Operational Research* 150, 2003, p. 496-515. 11. *D.Eckert, C.Klamler, J.Mitlohner and C.Schlotterer.* A distance-based comparison of basic voting rules. // *European Journal of Operations Research*, Vol. 12. *Даргейко, Л.Ф.* Технология многокритериальной оптимизации в системах поддержки принятия решений проблемы создания интеллектуальных систем поддержки принятия решений [Текст] / Л.Ф. Даргейко. – К., 1990. – С. 20-26. 13. *Моисеенко, В.В.* Системная оптимизация как обобщение оптимизации классической [Текст] / В.В. Моисеенко, В.В. Яцкевич // *Кибернетика и системный анализ.* – 1997. – №3. – С. 135-139.

Поступила в редколлегию 14.04.2010 г.

УДК 681.324.01

Задачі та концепції методів багатокритеріальних рішень в інтелектуальних системах / Г.І. Стопченко, І.А. Макрушан, С. В. Білан // *Біоніка інтелекту: наук.-техн. журнал.* – 2010. – № 1 (72). – С. 122–125.

Проведено аналіз типів постановок, багатокритеріальних задач прийняття рішень, що дозволяють визначити концептуальність математичних засобів прийняття рішень. Запропоновано ряд класифікаційних ознак, що дозволяють розширити існуючу модель прийняття рішень, і систему концепцій пошуку рішень.

Л. І. Бібліогр.: 17 найм.

UDC 681.324.01

Tasks and concepts of multicriterion decision-making methods in intelligent systems / G.I. Stopchenko, I.A. Makrushan, S.V. Bilan // *Bionics of Intelligence: Sci. Mag.* – 2010. – № 1 (72). – P. 122–125.

In the article present approaches to structuring methods of multicriterion decision-making are analysed. A range of classification criteria that allow to expand the present model of decision-making as well as the system of decisions searching concepts are proposed.

Fig. 1. Ref.: 17 items.

УДК 519.7: 004.8



ИСПОЛЬЗОВАНИЕ МЕТОДОВ ПЛАНИРОВАНИЯ ДЛЯ РЕШЕНИЯ ЗАДАЧ В ИНФОРМАЦИОННЫХ СИСТЕМАХ

А.А. Воскобойникова

Харьковский национальный университет радиоэлектроники, voskobjnikova@gmail.com

Рассмотрена проблема поиска решения по запросу к нескольким информационным системам, в основе которых лежит онтологическая структура представления знаний. Как способ решения заданной проблемы предложено использовать направление искусственного интеллекта – методы автоматического планирования. Приведено описание требований к содержанию и формальному описанию структуры плана, а также его положение в архитектуре интеграции информационных систем.

МЕНЕДЖМЕНТ ЗНАНИЙ, ИНТЕГРАЦИЯ ИНФОРМАЦИОННЫХ СИСТЕМ, ОНТОЛОГИЧЕСКАЯ СТРУКТУРА, ПЛАНИРОВАНИЕ, МЕТОДЫ ПОИСКА РЕШЕНИЙ

Введение

С развитием информационных технологий и World Wide Web актуальными стали вопросы о совместном использовании данных для различных пользователей (людей, систем, агентов, сервисов и так далее). Информационное пространство, развиваясь, стремится к однородности и общедоступности.

Одним из направлений на пути к совместному пользованию информацией является интеграция информационных систем, которая в последнее время стала актуальной задачей в свете мировой экономической обстановки, когда все чаще различные, в том числе и информационные, структуры объединяются в одну. Интеграция информационных систем происходит двумя путями, в зависимости от того, какими именно объектами оперируют интегрируемые системы. Первый путь интеграции – это интеграция данных. Этот подход применяется, когда основу бизнес-логики системы определяют структуры реляционных баз данных. Второй путь интеграции – интеграция знаний, применяется, когда бизнес-логика системы представлена в виде онтологической структуры. В то время, как первый путь интеграции предоставляет множество методов (консолидация, федерализация, распространение данных) и технологий (EAI, ETL, EIP) интеграции данных [1], второй путь оперирует лишь тремя подходами: интеграция онтологий, сопоставление (matching) и отображение (mapping) онтологических структур. Следует отметить, что интеграция онтологических структур приводит к образованию единой структуры из интегрируемых, что далеко

не всегда приемлемо для интегрируемых информационных систем, в основе которых лежат такие структуры. В то же время, отображение и сопоставление онтологических структур не решает вопроса интеграции и дальнейшего совместного использования таких структур (не предоставляет методов их совместной обработки), а решает задачи установления соответствия между отдельными объектами знаний системы [2, 3].

1. Проблема поиска решения задачи в интегрируемых онтологических структурах

Решение проблемы интеграции информационных систем, в основе которых лежит онтологическая структура, предложено в [4]. Суть подхода заключается в создании ядра интеграции, которое содержит в себе знания о взаимодействии интегрируемых систем и об интерфейсах общения с ними. Предложенная модель интеграции (рис. 1) позволяет решать проблемы реализации обращений с запросами от одной системы к другой посредством применения методов отображения и сопоставления онтологических структур. Однако остается проблема решения задач в рамках интегрируемых систем, которые требуют создания последовательности запросов к различным системам для получения конечного решения.

Рассмотрим ситуацию, когда реализуется интеграция нескольких систем S_i , где $i = \{min\ i = 2, max\ i = n\}$. Для ответа по запросу в такой архитектуре необходима реализация следующей последовательности действий:

1) формирование запроса в терминах системы, которая его посылает;

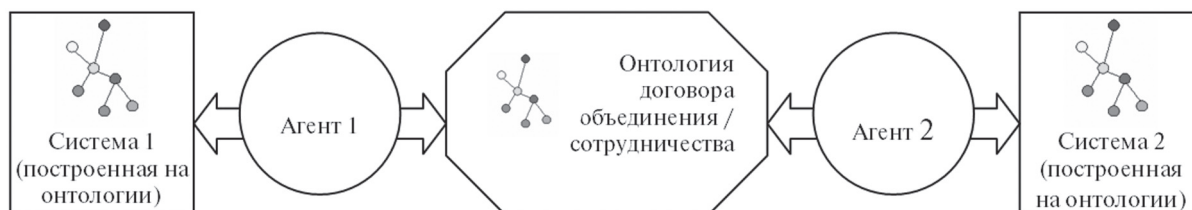


Рис. 1. Схема архитектуры интеграции интеллектуальных информационных систем

2) отправка запроса в ядро интеграции;

3) реализация разбиения запроса на подзадачи и формирование запросов соответственно каждой выявленной подзадаче;

4) определение системы, в которую следует перенаправить запрос, и представление его в терминах этой системы;

5) обработка решения каждой подзадачи и возвращение ответа системе, которая направила запрос.

Таким образом, при формировании запроса, который потребует создания последовательности других запросов для решения поставленной задачи, из некоторой системы S_i в архитектуре интегрированных систем, потребуется провести анализ $n-1$ систем для каждой k -ой подзадачи. Учитывая вычислительную сложность обработки онтологических структур, такая процедура поиска ответа по запросу потребует больших затрат ресурсов как временных, так и вычислительных.

Поэтому, предполагая, что множество сложных запросов, которые следует разбивать на подзадачи, конечно, целесообразно было бы хранить процедуру нахождения решения и при каждом запросе не проводить поиск ответа заново, а использовать готовый шаблон решения. Так же правомерно будет и утверждение о том, что часть шаблона поиска решения, которая содержит в себе некоторые подзадачи, может являться, в свою очередь, самостоятельным шаблоном поиска ответа на запрос, и комбинация нескольких шаблонов также может образовать новый шаблон нахождения ответа. Учитывая подобную специфику поиска решения, становится возможным применение планирования как процесса формирования шаблона.

2. Планирование как метод поиска решения

Планированием называется процесс выработки последовательности действий, позволяющих достичь цели. Существует классическое и неклассическое планирование. Первое оперирует наблюдаемыми, детерминированными, конечными, статическими, дискретными средами. Второе же – частично наблюдаемыми или стохастическими вариантами сред. Учитывая то, что главным свойством информационной системы, которая основана на онтологической структуре представления знаний, является возможность ее изменения в зависимости от изменений в предметной области и задач системы, рассматриваемая среда является стохастической.

Существует несколько методов планирования для стохастических сред, которые рассматриваются в рамках искусственного интеллекта [5]. Это:

1) планирование иерархической сети задач – первоначальный план, описывающий задачу, рассматривается как описание задачи на очень высо-

ком уровне абстракции. Выполнение такого плана происходит путем декомпозиции действий, когда действие высокого уровня сводится к частично упорядоченному набору действий низкого уровня;

2) планирование в недетерминированной среде – это возможность составления такого плана, который будет применим во всех возможных обстоятельствах и будет учитывать возможность появления неполных данных;

3) условное планирование – это составление плана с учетом неопределенности путем проверки того, что фактически происходит в среде при выполнении заранее заданных пунктов плана.

Рассматриваемая в работе среда – набор интегрированных информационных систем, основанных на онтологической структуре представления знаний, по сути, является полностью наблюдаемой, не содержащей противоречивой информации, однако недетерминированной. Поэтому примирительно к такой среде целесообразно будет применять, как один из методов планирования, задачи последовательного принятия решений.

В целом механизмы планирования позволяют проводить конкретизацию действий по нахождению решения задачи, учитывая последовательность действий, ограничения на их выполнение, временные ограничения и ограничения на использование ресурсов. Таким образом становится возможным утверждать, что при создании плана возникает некоторое знание о процедуре нахождения решения задачи, а также всех альтернативных процедурах поиска решения в условиях недоступности основной. Следовательно, планирование можно рассматривать как часть менеджмента знаний, что делает его мощным инструментом для решения задач в информационных системах, в основе которых лежит онтологическая структура представления знаний о предметной области.

3. Положение плана в архитектуре интеграции

Исходя из представленной в пункте 1 данной статьи процедуры поиска ответа на запрос, шаблоны планов с их формальным описанием должны быть расположены в ядре интеграции. Таким образом, единожды сформированный план, части плана или комбинация планов могут быть использованы любой из интегрируемых систем.

Предполагается рассматривать множество сформированных планов как множество экземпляров объекта онтологии интеграции, которая будет описывать механизмы взаимодействия интегрируемых систем. Реализация выполнения плана должна осуществляется посредством агентных технологий, реализованных на платформе JADE, которая предоставляет удобные интерфейсы как для создания интеллектуальных агентов, так и для обработки онтологических структур.

4. Требования к формальному описанию плана

Рассмотрим планирование с частичным упорядочением действий (так как полного упорядочения любого такого плана можно достичь при помощи процедуры линеаризации). Формально такой план состоит из следующих компонентов: множество действий, множество ограничений упорядочения, множество причинных связей, множество открытых предусловий. Причем, последние два множества определяют то, каким образом план может быть дополнен.

План называется согласованным, если он не имеет циклов в ограничениях упорядочения и конфликтов между причинными связями.

Также, в связи с тем, что описание плана должно быть реализовано в терминах онтологии, определяются следующие требования к его компонентам [6]:

1) начальное состояние среды, S_0 – описывает состояние среды в начальный момент планирования;

2) цель, G – описывает целевое состояние It describes the desired state of the world we want to achieve through a planning process;

3) множество подзадач планирования, $PT = \{pt_1, \dots, pt_n\}$ – набор подзадач планирования, которые определяют промежуточные цели, которые должны быть достигнуты в процессе планирования, для достижения общей цели задачи планирования;

4) действия – для каждой подзадачи pt_i существует конечный набор действий $A_i = \{a_{i1}, \dots, a_{ik}\}$, который должен быть выполнен, чтобы цель подзадачи pt_i была достигнута;

5) агенты, $AG = \{ag_1, \dots, ag_m\}$ – набор интеллектуальных агентов, которые реализуют достижение целей плана посредством выполнения задач;

6) параметры, $PA = \{pa_1, \dots, pa_l\}$ – параметры, которые применяются как указатели мета-уровня на объекты-домены, которые релевантны процессу планирования;

7) период времени, TH – промежуток времени, обязательный для выполнения плана;

8) ограничения, $C = \{c_1, \dots, c_j\}$ – набор ограничений, которые должны быть соблюдены во время выполнения плана;

9) множество настроек, $PR = \{pr_1, \dots, pr_o\}$ – набор критериев, которые позволяют ранжировать планы. Эти критерии важны для осуществления оптимизации процесса планирования, а также для процесса приобретения знаний, и помогают предотвратить использование взаимно невыполнимых планов. Эти критерии принято называть «мягкими ограничениями (soft constraint)» во многих подходах проектирования и планирования;

10) функция стоимости, Sf – функция, которая реализует общий механизм сравнения «стоимости» выполнения альтернативных планов;

11) критерии решения, SOL – механизмы отображения плана P на множество $\{True, False\}$, которые определяют, является ли план-кандидат решением задачи. Критерии устанавливают, является ли план P законченным и валидным;

12) модель плана, $P = \{p_1, \dots, p_q\}$ – план-кандидат, который определяется парой $\langle pt_i, ag_j \rangle$, где pt_i – это подзадача плана, а ag_j – агент, способный выполнить соответствующие действия для достижения цели задачи pt_i .

Не все из заданных параметров являются обязательными. Но потенциально они дают возможность поддерживать выразительную мощь онтологических структур для представления знаний в информационных системах. Таким образом, используя подобное описание плана, становится возможным реализовать процедуру поиска ответа на сложный запрос к множеству информационных систем, в основе которых лежит онтологическая структура представления знаний.

5. Методы и технологии реализации и описания плана

Основываясь на характеристиках рассматриваемой среды – онтологических структурах, которые являются основой информационных систем, и характеристиках методов автоматического планирования, можно сделать вывод, что оптимальным методом планирования для описанной проблемы является метод планирования иерархической сети задач с последовательным принятием решений.

Планирование иерархической сети задач основано на сетях HNT (Hierarchical Task Network) [5]. Как было отмечено в разделе 2, такой метод планирования предусматривает декомпозицию первоначальной задачи, которая имеет наиболее общее абстрактное описание. Происходит так называемая процедура декомпозиции действий, которая заканчивается, когда достигнуты примитивные действия. Примитивные действия – это действия, которые могут быть выполнены агентом автоматически. Декомпозиция действий, согласно этому методу, проходит последовательно, за счет чего реализуется процесс уточнения и конкретизации деятельности по достижению цели, что в данном контексте эквивалентно получению нового знания.

Следует отметить, что при постановке конъюнктивных целей применяется гибридный подход к планированию, в котором декомпозиции действий используются как уточнения плана в планировании с частичным упорядочением в дополнение к стандартным операциям определения открытых условий и разрешения конфликтов путем введения ограничений упорядочения [5].

Применяя методы последовательного принятия решений, в планировании становится возможным выбор наиболее оптимальной стратегии – страте-

гии, которая принесет максимальную полезность в результате выполнения плана.

Реализация выполнения процедуры построения плана наиболее эффективно будет представлена посредством создания интеллектуальных агентов. Этот инструмент был разработан специально для решения задач поиска решений проблем методами искусственного интеллекта.

Само представление формального описания плана целесообразно реализовывать средствами Semantic Web, так как именно его технологии имеют максимальную выразительность, которая позволит полно задать формальную структуру плана.

Одними из наиболее оптимальных технологий являются OWL (Ontology Web Language) и RDF (Resource Description Framework).

Основная часть формального описания — это создание системы ограничений. Для этого OWL имеет мощный инструмент. Существует возможность задавать ограничения как на значения параметров (*owl:Restriction*), так и на классы [7]. Существуют ограничения значения свойств: *owl:allValuesFrom*; *owl:someValuesFrom*; *owl:hasValue*. Также существует три типа ограничений количества элементов: *owl:maxCardinality*; *owl:minCardinality*; *owl:cardinality*.

Посредством ограничений *rdfs:domain* и *rdfs:range* возможно задавать классы экземпляров объектов с заданными свойствами.

Также в рамках развития технологии OWL представлен стандарт OWL-S (Semantic Markup for Web Services), который поддерживает взаимодействие между онтологическими структурами и семантическими web-сервисами.

Выводы

В данной работе была рассмотрена проблема поиска ответа на запрос в системе, которая содержит несколько онтологических структур. Рассматривалась система, полученная в результате интеграции нескольких информационных систем согласно методу, предложенному в [4].

Как решение проблемы было предложено использовать технологию автоматического планирования, которая является релевантным механизмом для реализации поиска решения в заданных условиях недетерминированной среды. Проведен анализ методов автоматического планирования, на основе которого выявлено, что метод планирования иерархической сети задач с последовательным принятием решений является наиболее оптимальным.

Определено место размещения шаблонов планов в архитектуре интеграции информационных систем — это ядро интеграции.

Для реализации процедуры планирования выбрана технология создания интеллектуальных агентов и платформа для ее реализации — JADE. В

качестве механизмов формального описания шаблонов процессов планирования предложено использовать стандарты W3C — OWL и RDF.

Список литературы. 1. *Воскобойникова, А.А.* Интеграция данных и знаний в информационной системе [Текст] / А.А. Воскобойникова: сб. наук. пр. — Т. 3 / ПП Вишневский В.С. — Херсон: ISDMCI, 2008. — С. 94-97. 2. *J. Euzenat, P. Shvaiko* Tutorial on Schema and Ontology Matching [Электронный ресурс] / ESWC'05. — Режим доступа: [www / URL: http://dit.unitn.it/~accord/Presentations/ESWC'05-MatchingHandOuts.pdf](http://dit.unitn.it/~accord/Presentations/ESWC'05-MatchingHandOuts.pdf) — 29.05.2005 г. — Загл. с экрана. 3. *Namyoun Choi, Il-Yeol Song, and Hyoil Han* A Survey on Ontology Mapping, Department of Information Engineering and Computer Science [Электронный ресурс] / UNIVERSITY OF TRENTO. — Режим доступа: [www / URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.107.4316&rep=rep1&type=pdf](http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.107.4316&rep=rep1&type=pdf) — 09.2006 г. — Загл. с экрана. 4. *Воскобойникова, А.А.* Разработка архитектуры интеграции нескольких информационных систем [Текст] / А.А. Воскобойникова // Восточно-Европейский журнал передовых технологий. — 2009. — №4/3(40). — с. 12-15. 5. *Рассел, С.* Искусственный интеллект: современный подход, 2-ое изд. [Текст] / С. Рассел, П. Норвиг: пер. с англ. — М.: Издательский дом «Вильямс», 2007. — 1408 с. 6. *D. Rajpathak, E. Motta* An Ontological Formalization of the Planning Task [Электронный ресурс] / Knowledge Media Institute, The Open University. — Режим доступа: [www / URL: http://eprints.aktors.org/504/01/Fois2004_camera-ready-final.pdf](http://eprints.aktors.org/504/01/Fois2004_camera-ready-final.pdf) — 2004 г. — Загл. с экрана. 7. OWL Web Ontology Language Reference [Электронный ресурс] / World Wide Web Consortium — Режим доступа: [www / URL: http://www.w3.org/TR/owl-ref/](http://www.w3.org/TR/owl-ref/) — 10.02.2004 г. — Загл. с экрана.

Поступила в редакцию 15.04.2010 г.

УДК 519.7: 004.8

Використання методів планування для вирішення задач в інформаційних системах / Г.А. Воскобойникова // Біоніка інтелекту: наук.-техн. журнал. — 2010. — № 1 (72). — С. 126–129.

В інформаційних технологіях зараз існує дві тенденції — інтелектуалізація та глобалізація. В роботі розглядається проблема пошуку вирішення задач у випадку, коли інформаційне середовище — це декілька інформаційних систем, які проінтегровано в одну. Кожна з заданих систем побудована на основі онтологічного підходу до представлення знань. У якості вирішення розглянутої проблеми пропонується застосування автоматичного планування, яке реалізується за допомогою інтелектуальних агентів.

Л. 1. Бібліогр.: 7 найм.

UDC 519.7: 004.8

Planning methods for solving problems in information systems / A.A. Voskoboynikova // Bionics of Intelligence: Sci. Mag. — 2010. — № 1 (72). — P. 126–129.

Intellectualization and globalization are the main trends in information technology today. In this paper the problem of finding solutions to the challenges when environment are a few information systems that integrate into one system, and all these systems are the ontology-based. As a solution to this problem are encouraged to apply automatic planning which implemented using intelligent agents.

Fig. 1. Ref.: 7 items.

ОБ АВТОРАХ

Авраменко Валерий Павлович	55	д-р техн. наук, профессор кафедры инженерной и компьютерной графики Харьковского национального университета радиоэлектроники
Баранов Алексей Васильевич	29	аспирант кафедры системотехники Харьковского национального университета радиоэлектроники
Бескорвайный Владимир Валентинович	50	д-р техн. наук, профессор кафедры системотехники Харьковского национального университета радиоэлектроники
Билан Светлана Викторовна	122	студентка кафедры информационных управляющих систем Харьковского национального университета радиоэлектроники
Ведмедь Андрей Григорьевич	33	аспирант факультета прикладной математики и менеджмента Харьковского национального университета радиоэлектроники
Вечирская Ирина Дмитриевна	83	канд. техн. наук, старший научный сотрудник кафедры программного обеспечения ЭВМ Харьковского национального университета радиоэлектроники
Виклюк Ярослав Игоревич	109	докторант, доцент, канд. физ.-мат. наук НУ «Львівська політехніка», г. Львов
Воскобойникова Анна Андреевна	126	аспирант кафедры искусственного интеллекта Харьковского национального университета радиоэлектроники
Гребенник Игорь Валериевич	29	д-р техн. наук, профессор кафедры системотехники Харьковского национального университета радиоэлектроники
Грицай Денис Васильевич	29	студент факультета компьютерных наук Харьковского национального университета радиоэлектроники
Гришко Андрей Александрович	18	аспирант кафедры электронных вычислительных машин Харьковского национального университета радиоэлектроники
Дубницкий Валерий Юрьевич	43	доцент кафедры высшей математики Харьковского института банковского дела Университета банковского дела
Збитнева Майя Вячеславовна	104	канд. техн. наук, доцент кафедры программного обеспечения ЭВМ Харьковского национального университета радиоэлектроники
Кобылин Анатолий Михайлович	43	доцент кафедры информационных технологий Харьковского института банковского дела Университета банковского дела
Кобылин Олег Анатольевич	43, 89	доцент кафедры информатики Харьковского национального университета радиоэлектроники
Кораблев Николай Михайлович	70	канд. техн. наук, доцент кафедры электронных вычислительных машин Харьковского национального университета радиоэлектроники
Кривчикова Анна Александровна	24	аспирантка кафедры информационных управляющих систем Харьковского национального университета радиоэлектроники
Крыгин Максим Юрьевич	75	научный сотрудник Украинского языково-информационного фонда, г. Киев

Крючковский Виктор Владимирович	61	канд. физ.-мат. наук, декан факультета международного менеджмента Херсонского национального технического университета
Левыкин Виктор Макарович	24	д-р техн. наук, профессор, заведующий кафедрой информационных управляющих систем Харьковского национального университета радиоэлектроники
Любченко Валентин Анатольевич	89	доцент кафедры информатики Харьковского национального университета радиоэлектроники
Макрушан Ирина Анатольевна	122	ассистент кафедры информационных управляющих систем Харьковского национального университета радиоэлектроники
Мальков Юрий Анатольевич	118	аспирант кафедры электронно-вычислительных машин Харьковского национального университета радиоэлектроники
Машталер Владимир Петрович	33	д-р техн. наук, профессор, декан факультета компьютерных наук Харьковского национального университета радиоэлектроники
Машталер Сергей Владимирович	65	канд. техн. наук, доцент кафедры информатики Харьковского национального университета радиоэлектроники
Оробинский Павел Андреевич	12	аспирант факультета прикладной математики и менеджмента Харьковского национального университета радиоэлектроники
Панкратов Александр Викторович	29	канд. техн. наук, научный сотрудник Института проблем машиностроения им. А.Н. Подгорного НАН Украины, г. Харьков
Пелешко Дмитрий Дмитриевич	98	канд. техн. наук, доцент кафедры АСУ Института компьютерных наук и информационных технологий Национального университета «Львівська політехніка»
Пискалова Ольга Александровна	61	ассистент кафедры системотехники Харьковского национального университета радиоэлектроники
Постульга Сергей Витальевич	65	соискатель кафедры информатики Харьковского национального университета радиоэлектроники
Путятин Евгений Петрович	12	д-р техн. наук, профессор, заведующий кафедрой информатики Харьковского национального университета радиоэлектроники
Рашкевич Юрий Михайлович	98	д-р техн. наук, профессор, проректор по научной работе и международным связям, заведующий кафедрой АСУ Института компьютерных наук и информационных технологий Национального университета «Львівська політехніка»
Русакова Наталья Евгеньевна	94	аспирантка кафедры программного обеспечения ЭВМ Харьковского национального университета радиоэлектроники
Сакало Евгений Сергеевич	33	аспирант факультета прикладной математики и менеджмента Харьковского национального университета радиоэлектроники
Селезнев Сергей Николаевич	24	аспирант кафедры информационных управляющих систем Харьковского национального университета радиоэлектроники

Соболева Елена Владимировна	50	аспирантка кафедры системотехники Харьковского национального университета радиоэлектроники
Стопченко Геннадий Иванович	122	канд. техн. наук, доцент кафедры информационных управляющих систем Харьковского национального университета радиоэлектроники
Субботин Сергей Александрович	38	канд. техн. наук, доцент кафедры программных средств Запорожского национального технического университета
Танянский Сергей Станиславович	118	канд. техн. наук, доцент кафедры электронно-вычислительных машин Харьковского национального университета радиоэлектроники
Ткаченко Владимир Филиппович	55	профессор, заведующий кафедрой инженерной и компьютерной графики Харьковского национального университета радиоэлектроники
Удовенко Сергей Григорьевич	18	д-р техн. наук, профессор кафедры электронных вычислительных машин Харьковского национального университета радиоэлектроники
Федорова Татьяна Николаевна	83	аспирантка кафедры программного обеспечения ЭВМ Харьковского национального университета радиоэлектроники
Филипская Диана Ивановна	61	младший научный сотрудник кафедры системотехники Харьковского национального университета радиоэлектроники
Фомичев Александр Александрович	70	аспирант кафедры электронных вычислительных машин Харьковского национального университета радиоэлектроники
Чалая Лариса Эрнестовна	18	канд. техн. наук, старший научный сотрудник, доцент кафедры искусственного интеллекта Харьковского национального университета радиоэлектроники
Чалый Сергей Федорович	24	д-р техн. наук, профессор кафедры информационных управляющих систем Харьковского национального университета радиоэлектроники
Четвериков Григорий Григорьевич	3, 83	д-р техн. наук, профессор кафедры программного обеспечения ЭВМ Харьковского национального университета радиоэлектроники
Чибирев Андрей Дмитриевич	55	аспирант кафедры инженерной и компьютерной графики Харьковского национального университета радиоэлектроники
Чугай Андрей Михайлович	29	канд. техн. наук, младший научный сотрудник Института проблем машиностроения им. А.Н. Подгорного НАН Украины, г. Харьков
Щербинин Константин Сергеевич	65	аспирант кафедры информатики Харьковского национального университета радиоэлектроники

ПРАВИЛА ОФОРМЛЕНИЯ РУКОПИСЕЙ ДЛЯ АВТОРОВ НАУЧНО-ТЕХНИЧЕСКОГО ЖУРНАЛА «БИОНИКА ИНТЕЛЛЕКТА»

Научно-технический журнал «Бионика интеллекта» принимает для опубликования написанные специально для него оригинальные рукописи, которые нигде раньше не издавались. Структура рукописи должна быть следующей: индекс УДК, заглавие, сведения об авторах, аннотация, ключевые слова, введение, основной текст статьи, выводы, список использованной литературы.

Согласно Постановлению ВАК Украины от 15.01.2003 №7-05/1 (Бюллетень ВАК, №1, 2003, с. 2) статья должна иметь следующие необходимые элементы: постановка проблемы в общем виде и ее связь с важными научными или практическими заданиями; анализ последних исследований, публикаций и выделение нерешенных ранее частей общей проблемы в данной области; формулирование целей и задач исследования; изложение основного материала исследований с полным обоснованием полученных научных результатов; выводы по данному исследованию и перспективы дальнейших исследований в данном направлении.

Статьи должны быть выполнены в редакторе Microsoft Word. Формат страницы — А4 (210 x 297 мм), поля: верхнее — 25 мм, нижнее — 20 мм, левое, правое — 17 мм. Количество колонок — 2, с интервалом между ними 5 мм, основной шрифт Times New Roman, кегль основного текста — 10 пунктов, междустрочный интервал — множитель (1,1), абзацный отступ — 6 мм. Объем рукописи — от 4 до 12 страниц (языки: русский, украинский, английский).

УДК печатается с первой строки, без отступа, выравнивание по левому краю.

Название рукописи печатается прописными буквами; шрифт прямой, полужирный, кегль 12.

Название разделов нумеруют арабскими цифрами, выделяют жирным шрифтом. Отступы для названия рукописи, инициалов и фамилий авторов, сведений об авторах, названий разделов, введения и выводов, списка литературы сверху — 6 пт, снизу — 3 пт.

Аннотация (на языке рукописи, абзац 4–10 строк, кегль 9) помещается в начале статьи и содержит информацию о результатах описанных исследований.

Ключевые слова (4–10 слов из текста статьи, которые с точки зрения информационного поиска несут смысловую нагрузку, на языке рукописи, через запятую в именительном падеже, кегль 9).

Рисунки и таблицы (черно-белые, контрастные) помещаются в текст после первой ссылки в виде встроенных объектов и раздельно нумеруются арабскими цифрами сквозной нумерацией, при наличии более одного объекта. Простые схемы, состоящие из 3–4 элементов, рисуют, используя вставку объекта Рисунок Microsoft Word. Более сложные выполняют в графических редакторах в виде черно-белых графических файлов форматов .tiff, .jpg, .wmf, .cdr с разрешением 300 dpi. Рисунки должны быть включены в текстовый файл, а также обязательно представлены отдельным файлом с соответствующим названием (например, ris_1.cdr).

Все элементы рисунка, включая надписи, должны быть сгруппированы. Все надписи в рисунках и таблицах должны быть выполнены шрифтом Times New Roman, кегль в рисунках — 10, в таблицах — 9. Табличный заголовок располагается справа над таблицей (кегель 9 пт).

Рисунок содержит подрисовочную центрированную подпись (вне рисунка), кегль 9, по центру, отступы сверху и снизу по 6 пт. Ширина рисунка должна соответствовать ширине колонки (или ширине страницы).

Формулы, символы, переменные должны быть набраны в редакторе формул MathType или Microsoft Equation. Формулы располагаются по центру и нумеруются при наличии ссылки на них в рукописи. Шрифт — Times New Roman. Высота переменной — 10 пунктов, индексов и подиндексов — 8 пт, основной математический символ — 12 (10) пт. Переменные, обозначенные латинскими буквами, набирают курсивом, греческие буквы, сокращения русских слов и цифры — прямым начертанием. Переменные, содержащиеся в тексте, также набирают в редакторе формул.

Небольшие формулы можно размещать не в отдельной строке, а прямо по тексту.

Список литературы включает опубликованные источники, на которые имеются ссылки в тексте, заключенные в квадратные скобки, печатается без абзацного отступа, кегль 9 пунктов, отступ сверху — 6 пт.

После списка литературы с отступами сверху — 6 пт указывается дата поступления статьи в редколлегию. Число и месяц задаются двузначными числами через точку. Размер шрифта — 9 пт, курсив, выравнивание по правому краю.

Рефераты (Times New Roman, кегль — 9 пунктов, 3–4 предложения) на украинском и английском языках. Реферат не должен дублировать текст аннотации.

Вместе с рукописью (на листах белой бумаги формата А4 плотностью 80–90 г/м², напечатанной на лазерном принтере, в 2-х экземплярах) необходимо подать следующие документы:

1. Заявку утвержденного образца, подписанную всеми авторами:

«Прошу принять статью ... (указать фамилии авторов, название статьи) объемом ... страниц (указать количество страниц) для опубликования в научно-техническом журнале «Бионика интеллекта». Оплату гарантируем.

Сведения об авторах: (указать фамилию, имя, отчество каждого автора, место работы, ученую степень, научное звание, контактный телефон, почтовый и электронный адреса).

Подписи авторов».

2. Акт экспертизы о возможности опубликования материалов в открытой печати.

3. Рецензию, подписанную доктором наук.

4. Сведения об авторах.

5. Электронный вариант рукописи, реферата и сведений об авторах.

6. Оплату за публикацию.

Наукове видання

БІОНІКА ІНТЕЛЕКТУ
інформація, мова, інтелект

Науково-технічний журнал

№ 1 (72)

2010

Головний редактор — *М. Ф. БОНДАРЕНКО*

Відповідальний редактор — *Ю. П. Шабанов-Кушнарєнко*

Заступник відповідального редактора — *Г. Г. Четвериков*

Відповідальний секретар — *І. Д. Вечірська*

Коректор — *Л. М. Денісова*

Комп'ютерна верстка — *О. Б. Ісаєва*

Рекомендовано Вченою Радою
Харківського національного університету радіоелектроніки
(протокол № 62 від 23.04.2010)

Адреса редакції:

Україна, 61166, Харків-166, просп. Леніна, 14,
Харківський національний університет радіоелектроніки, к. 127, 285
тел. 702-14-77, факс 702-10-13,
e-mail: bionics@kture.kharkov.ua

Підписано до друку 29.04.2010. Формат 60 x 84 ¹/₈. Друк ризографічний.
Папір офсетний. Гарнітура Newton. Умов. друк. арк. 15,6. Обл.-вид. арк. 15,0.
Тираж 300 прим. Зам. № .

Надруковано в навчально-науковому видавничо-поліграфічному центрі ХНУРЕ
61166, Харків-166, просп. Леніна, 14