MACHINE LEARNING IN CLASSIFICATION TASKS

Калайчев Г.В.
Науковий керівник – д-р фіз.-мат. наук, доц. Сидоров М.В.
Харківський національний університет радіоелектроніки
61166, Харків, просп. Науки, 14, каф. прикладної математики,
тел. (057) 702-14-36, e-mail: heorhii.kalaichev@nure.ua

The main goal of this work is to show the ways to use machine learning algorithms to solve classification tasks. One of the most efficient algorithms is Gradient Boosting (XGB Classier). This is a method which is usually used in competitions because of his speed and opportunity to work with big amount of data.

Останнім часом алгоритми машинного навчання знаходять все більш широке використання в світі технологій. Їх використовують у медицині для розпізнавання пухлин на знімках томографа, для прогнозування діагнозу хворого, скільки він проведе часу в лікарні, розпізнавання авто на дорозі, розпізнавання пішоходів тощо.

1. Методи машинного навчання у задачах класифікації. Одним з найстаріших алгоритмів є лінійний алгоритм класифікації – логістична регресія [1]. Розглянемо алгоритм класифікації – метод опорних векторів (SVM) [2]. Задача алгоритму SVM – максимізувати розділення, що визначається, як відстань між гіперплощиною, що розділяє два класи, і найближчими до цієї гіперплощини тренувальними зразками, які називаються опорними векторами. Якщо дані не є лінійно роздільними, використовують ядерний SVM. Основною ідеєю цього методу є те, що тренувальні дані переводяться у простір ознак більш високої вимірності використовуючи відображення $\phi(\square)$, потім тренують лінійну модель SVM в новому просторі ознак. Деревом рішень [3] називається кореневе дерево, в якому кожен внутрішній вузол (не лист) помічений ознакою і кожне ребро, що виходить із внутрішнього вузла, подає можливі відповіді на запитання, асоційоване з цією вершиною. Кожен листок має мітку класу, до якої відноситься об’єкт. При використанні дерева рішень можна отримати перенавчання, оскільки може утворитися дуже глибоке дерево з багатьма вузлами, і тоді модель не навчиться розпізнавати класи, а просто «завчить» відповіді та на тестових даних ми отримаємо поганий результат.

2. Алгоритм XGBoost Classifier в задачах класифікації. Зазвичай у задачах класифікації, як у всіх задачах, які відносяться до задач навчання з вчителем, дані складаються з об’єктів, які мають ознаки, що включають у себе мітки класів. Дані подаються у вигляді матриці $X_{n\times m}$. При розв’язанні задачі класифікації необхідно навчити модель на основі ознак в даних передбачувати належність об’єкта до одного з класів – надати об’єкту мітку класу. Для роботи з великими об’ємами даних бажано використовувати алгоритми класифікації, які досить швидко з ними працюють і при цьому не втрачають у стійкості та точності. Найбільш популярними є алгоритми на основі дерев, а саме XGBoost Classifier, як один з найпотужніших алгоритмів класифікації. XGBoost первинно було

розпочато, як дослідницький проект Тенці Чжена у складі групи Спільноти Глибинного Машинного навчання у 2014 р. [3]. Ця технологія швидко набула популярності через перемоги у багатьох змаганнях з машинного навчання.

Основною ідеєю XGBoost Classifier є те, що необхідно побудувати композицію з декількох базових алгоритмів. У нашому випадку кожен базовий алгоритм – це J-термінальне дерево. Кожне дерево має адитивну форму:

$$h(x;\{b_j,R_j\}_1^J)=\sum_{j=1}^{J} b_j I(x\in R_j),$$

де $\{R_j\}_1^J$ – різні набори даних, які не перетинаються один з одним та разом покривають увесь набір даних, що було подано у дерево. Ці набори даних знаходяться в термінальних вершинах дерева.

Отже, алгоритм XGBoost Classifier складається з наступних кроків.

Крок 1. Обираємо слабку модель $F_0(x)=\frac{1}{2}\log_2\frac{1+\bar{y}}{1-\bar{y}}$, де $\bar{y}$ – це середнє значення вектору відповідей.

Крок 2. Для m від 1 до M, де M – кількість дерев у класифікаторі виконуємо наступні дії:

– отримуємо передбачення алгоритму, враховуючи результати попереднього алгоритму;

– оновлюємо дані у кожній окремій термінальній вершині;

– доповнюємо попередню модель за формулою:

$$F_m(x)=F_{m-1}(x)+\sum_{j=1}^{J}\gamma_{jm} I(x\in R_{jm}).$$

Результати використання описаних алгоритмів показали, що найкращим та найстабільнішим алгоритмом є XGBoost Classifier, оскільки він є більш стійким до викидів та перенавчання. Але, при роботі з меншим об’єм даних (матриця об’єкти-ознаки має менше ознак), інші алгоритми показують досить високі результати.

Список використаних джерел:

1. Орельен Ж. Прикладное машинное обучение с помощью Scikit-Learn и TensorFlow. Концепции, инструменты и техники для создания интеллектуаль-ных систем. Москва : Вильямс, 2018. 688 с.

2. Speech emotion recognition using support vector machine / M. Jain, S. Narayan, P. Balaji [and other] // arXiv preprint arXiv:2002.07590. 2020. 6 P.

3. Wang C., Deng C., Wang S. Imbalance-XGBoost: leveraging weighted and focal losses for binary label-imbalanced classification with XGBoost // Pattern Recognition Letters. 2020. № 2. P. 1-11.