

Е.В. БОДЯНСКИЙ, О.В. ЗАПОРОЖЕЦ,
Т.В. ПУТЯТИНА, О.Е. РАГУЛИНА

НЕЙРОСЕТЕВАЯ МОДЕЛЬ ФАКТОРНОГО АНАЛИЗА

Важнейшей проблемой, возникающей при обработке больших массивов наблюдений, является задача сжатия данных в целях выявления наиболее существенной информации и латентных факторов, определяющих суть изучаемого явления, что в конечном итоге преследует цель сделать исходные данные обозримыми, достаточно просто интерпретируемыми для исследователя.

Одним из наиболее эффективных подходов к решению этой проблемы является аппарат факторного анализа [1-3], применяющийся в задачах обработки эмпирических данных в самых различных областях: психологии, социологии, экономике, технике, медицине, криминалистике – там, где получение информации связано с массовыми обследованиями, а возможность однозначной интерпретации данных отсутствует.

Основная идея факторного анализа, предполагающая наличие не известных заранее латентных факторов, состоит в следующем: наблюдая большое число измеряемых параметров-факторов, которые в основном определяют их поведение, или иначе: зная большое число функций измеряемых параметров, определить соответствующие значения общих для всех функций аргументов-факторов и установить вид этих функций.

Исходной информацией для факторного анализа является $(t \times n)$ -матрица наблюдений

$$X_t = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ x_{i1} & x_{i2} & \dots & x_{in} \\ \vdots & \vdots & \vdots & \vdots \\ x_{t1} & x_{t2} & \dots & x_{tn} \end{pmatrix} = \begin{pmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_i^T \\ \vdots \\ x_t^T \end{pmatrix} \begin{matrix} i = 1, 2, \dots, t, \\ k = 1, 2, \dots, n, \end{matrix} \quad (1)$$

образованная массивом из t n -мерных векторов $x_i = (x_{i1}, x_{i2}, \dots, x_{in})^T$, и ее автокорреляционная $(n \times n)$ -матрица

$$R_t = \frac{1}{t-1} \sum_{i=1}^t (x_i - \bar{x}_i)(x_i - \bar{x}_i)^T = \frac{1}{t-1} \sum_{i=1}^t \tilde{x}_i \tilde{x}_i^T, \quad (2)$$

где $\bar{x}_i = \frac{1}{l} \sum_{i=1}^l x_i$, $\tilde{x}_i = x_i - \bar{x}_i$ — центрированные относительно среднего

исходные данные.

Одним из наиболее распространенных и эффективных методов нахождения факторов является метод главных компонент, или компонентный анализ, широко применяющийся в задачах сжатия данных, распознавания образов, кодирования обработки изображений, спектрального анализа и т.д., известный в теории распознавания образов еще как преобразование Карунена-Лозва.

Задача компонентного анализа состоит в проецировании входных данных из исходного n -мерного пространства в m -мерное ($m < n$) выходное и сводится к нахождению системы w^1, w^2, \dots, w^m ортонормальных собственных векторов матрицы R_l таких, что $w^1 = (w_1^1, w_2^1, \dots, w_n^1)^T$ соответствует наибольшему собственному значению λ_1 матрицы R_l , $w^2 = (w_1^2, w_2^2, w_3^2, \dots, w_n^2)^T$ второму по величине собственному значению λ_2 и т.д. Другими словами, речь идет о нахождении решения матричного уравнения

$$(R_l - \lambda_k I)w^k = 0 \quad (3)$$

такого, что $\lambda_1 \geq \lambda_2 \geq \dots \lambda_n \geq 0$ и $\|w^k\|^2 = 1$. Здесь I — $(n \times n)$ -единичная матрица, а m определяется числом собственных значений, удовлетворяющих, например, условию $\lambda_k \geq 1$ [1].

Таким образом, в алгебраических терминах решение факторной задачи тесно связано с проблемой собственных значений и нахождением ранга корреляционной матрицы; в геометрическом смысле — это задача перехода в пространство более низкой размерности с минимальной потерей информации; в статистическом смысле — это задача последовательного определения множества собственных векторов w^1, w^2, \dots, w^m путем последовательной максимизации каждого из частных функционалов, входящих в глобальный критерий

$$J = \frac{1}{l} \sum_{k=1}^m \sum_{i=1}^l (\tilde{x}_i^T w^k)^2 \quad (4)$$

при ограничениях $w^{kT} w^l = 0$ при $k \neq l$, $w^{kT} w^k = 1$.

Первая главная компонента этого множества вектор w^1 находится путем максимизации выражения

$$J^1 = \frac{1}{l} \sum_{i=1}^l (x_i^T w^1)^2 \quad (5)$$

с помощью метода неопределенных множителей Лагранжа. Далее из каждого вектора x_i вычитается его проекция на первую главную компоненту и вычисляется первая главная компонента остатков, которая является второй главной компонентой исходных данных и ортогональна первой. Третья главная компонента вычисляется путем проекции каждого входного вектора на первую и вторую главную компоненты, вычитания этой проекции из каждого x_i и нахождения первой главной компоненты полученных остатков, которая и есть третья главная компонента исходных данных. Остальные главные компоненты вычисляются рекурсивно согласно описанной стратегии.

Существует достаточно развитое программное обеспечение для решения факторной проблемы, объединяемое общим недостатком: необходимостью априорного задания матрицы X_t фиксированной размерности. Если же данные поступают последовательно в реальном масштабе времени, стандартные процедуры факторного анализа становятся неработоспособными.

В связи с этим представляется целесообразным использование рекуррентных процедур для нахождения в реальном времени собственных векторов матрицы R путем последовательной обработки поступающих входных сигналов $x_1, x_2, \dots, x_t, x_{t+1}, \dots$, без вычисления самой корреляционной матрицы. Такой подход может оказаться весьма эффективным при работе с объектами, характеризующимися дрейфом параметров, а его развитие связано, прежде всего, с адаптивными и нейросетевыми технологиями [4-7].

Для нахождения первой главной компоненты Оя предложил [8] алгоритм самообучения линейного нейрона типа Адалина Уидроу [4, 6], приведенного на рис. 1. Для предварительно центрированных данных алгоритм Оя имеет вид

$$\begin{cases} w_{t+1}^1 = w_t^1 + \gamma_{t+1} (\tilde{x}_{t+1} - \varphi_{t+1}^1 w_t^1) \varphi_{t+1}^1, \\ \varphi_{t+1}^1 = \tilde{x}_{t+1}^T w_t^1, w_0^1 \neq 0, \varphi_1^1 = \tilde{x}_1^T w_0^1, \end{cases} \quad (6)$$

где γ_{t+1} – параметр шага поиска, выбираемый достаточно малым для устойчивой работы алгоритма.

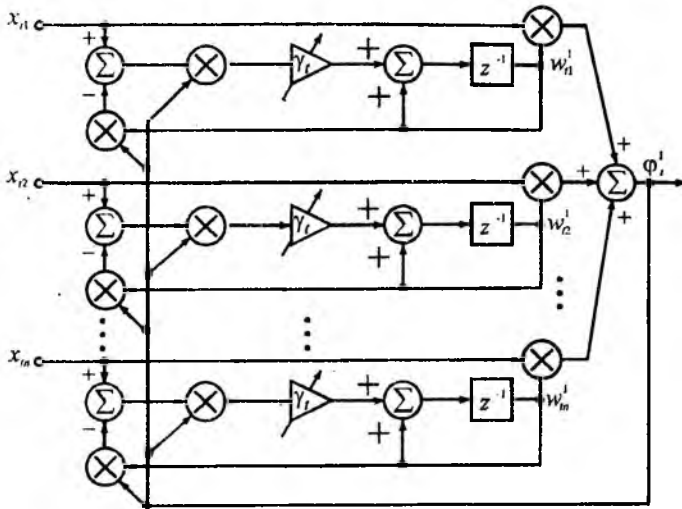


Рис.1 Нейрон Оя

В [9] была доказана сходимость этого алгоритма в предположении, что шаг поиска выбирается в соответствии с условиями Дворецкого, в связи с чем представляется целесообразным рассчитывать этот параметр согласно алгоритму [10]

$$\gamma_{t+1} = \gamma_t^{-1}; \quad r_{t+1} = \alpha r_t + \|\tilde{x}_{t+1}\|^2, \quad 0 \leq \alpha \leq 1, \quad (7)$$

где α – параметр, определяющий компромисс между следящими и сглаживающими свойствами процедуры (6). Несложно видеть, что при $\alpha = 0$ алгоритм Оя приближается к алгоритму обучения Уидроу-Хоффа [6], а при $\alpha = 1$ является процедурой стохастической аппроксимации типа Гудвина-Рэмеджа-Кейнеса [4]. Доказано также, что алгоритм Оя обеспечивает нормирование вектора w_i^1 так, как это показано на рис. 2

($\|w_i^1\|^2 = 1$), сам вектор w_i^1 является собственным вектором матрицы R_t ,

а выходной сигнал нейрона Оя ϕ_i^1 характеризуется максимально возможной дисперсией, т.е. содержит максимум информации о входном сигнале \tilde{x}_t .

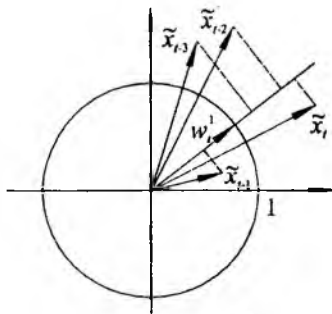


Рис. 2. Множество векторов и первая главная компонента

Для нахождения m главных компонент в реальном времени можно воспользоваться идеей рекурсивного их вычисления по описанной выше стратегии и обобщенным алгоритмом Хебба-Сэнгера [4,12], который в модифицированной форме имеет вид:

$$\begin{cases} w_{t+1}^l = w_t^l + \gamma_{t+1} \hat{x}_{t+1}^l \Phi_{t+1}^l, \\ \hat{x}_{t+1}^l = \hat{x}_{t+1}^{l-1} - \Phi_{t+1}^l w_t^l, \\ \hat{x}_{t+1}^0 = \tilde{x}_{t+1}, l = 1, 2, \dots, m; m < n, \\ \gamma_{t+1} = r_{t+1}^{-1}, r_{t+1} = \alpha r_t + \|\tilde{x}_{t+1}\|^2, 0 \leq \alpha \leq 1. \end{cases} \quad (8)$$

Видно, что первая главная компонента вычисляется с помощью нейрона Оя, далее проекции входных векторов на w^1 вычитаются из входов и остатки обрабатываются вторым нейроном и т.д.

На рис. 3 приведена схема искусственной нейронной сети, набранной из нейронов Оя и реализующей модифицированный обобщенный алгоритм Хебба-Сэнгера. Во входном слое сети проводится центрирование векторов x_t согласно алгоритму

$$\begin{cases} \tilde{x}_{t+1} = x_{t+1} - \bar{x}_{t+1}, \\ \bar{x}_{t+1} = \bar{x}_t + \frac{1}{t+1} (x_{t+1} - \bar{x}_t). \end{cases} \quad (9)$$

Далее сигналы \tilde{x}_t обрабатываются набором из m нейронов Оя и, наконец, выходной слой образован линейными элементами с зоной нечувствительности δ , позволяющими выделить наиболее информативные сигналы Φ_t^l и отсеять шум.

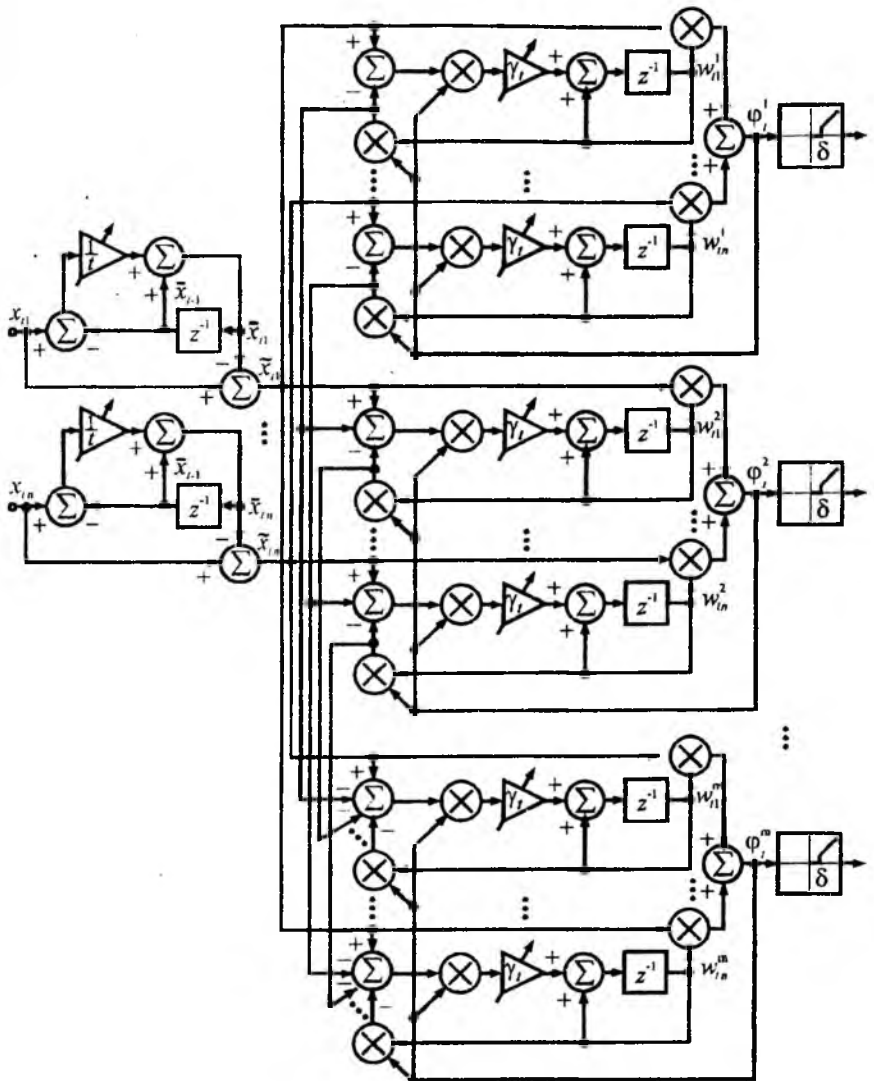


Рис. 3. Нейросетевая модель факторного анализа

Рассмотренная нейросетевая модель факторного анализа позволяет в реальном времени обрабатывать большие массивы информации и обеспечивать ее сжатие с требуемой степенью точности.

- Список литературы:** 1. Иберла К. Факторный анализ. М.: Статистика, 1980. 398 с.
2. Лоули Д., Максвелл А. Факторный анализ как статистический метод. М.: Мир, 1967. 144 с.
3. Браверман Э.М., Мучник Б. Структурные методы обработки эмпирических данных. М.: Наука, 1983. 464 с. 4. *Sichocki A., Unbehauen R. Neural Networks for Optimization and Signal Processing.* Stuttgart: Teubner, 1993. 526 p. 5. *Advances in Intelligent Control / Ed. by C.J. Harris.* London: Taylor and Francis. 1994. 373 p. 6. *Rojas R. Neural Networks. A Systematic Introduction.* Berlin: Springer-Verlag. 1995. 238 p. 7. *Bishop C.M. Neural Networks for Pattern Recognition.* Oxford: Clarendon Press. 1995. 482 p. 8. *Oja E. A simplified neuron model as a principal component analyzer // J. of Math. Biology.* 1982. №18. P. 267 — 273. 9. *Chen T., Hua Y., Yan W.-Y. Global convergence of Oja's subspace algorithm for principal component extraction // IEEE Trans. on Neural Networks.* 1998. №9. P. 58-67. 10. Бодянский Е.В., Плисс И.П., Соловьева Т.В. Многошаговые оптимальные упреждители многомерных нестационарных стохастических процессов // Докл. АН УССР. 1986. А. № 12. С. 47-49. 11. *Goodwin G. C., Ramadge P.J., Caines P.E. A globally convergent adaptive predictor // Automatica.* 1981. №17. P. 135-140. 12. *Sanger T. Optimal unsupervised learning in a single-layer linear feedforward neural network // Neural Networks.* 1989. №2. P. 459-473.

Поступила в редколлегию 03.06.99