ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

5(201)
2013

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Издается с ноября 1995 г.

УЧРЕДИТЕЛЬ
Издательство "Новые технологии"

СОДЕРЖАНИЕ

Информация о журнале доступна по сети Internet по адресу **http://novtex.ru/IT.**
Журнал включен в систему Российского индекса научного цитирования.
Журнал входит в Перечень научных журналов, в которых по рекомендации ВАК РФ должны быть опубликованы научные результаты диссертаций на соискание ученой степени доктора и кандидата наук.

УДК 519.246

Б. Г. Кухаренко, канд. физ.-мат. наук,
ст. науч. сотр., вед. науч. сотр.,
Институт машиноведения РАН, г. Москва,
e-mail: kukharenko@imash.ru,
Д. И. Пономарев, аспирант,
Московский физико-технический институт (ГУ),
e-mail: ponomarev-102@mail.ru

Анализ независимых компонент потока векторов для сокращения размерности пространства при кластеризации векторов в целях абстракции данных

Для обеспечения устойчивости кластеризации векторов на заданное число кластеров традиционно используется Анализ главных компонент. Показано, что при Анализе главных компонент потока векторов абстракция данных в результате кластеризации временны́х сечений главных компонент только приближенно представляет абстракцию данных исходного потока векторов. Однако при сокращении размерности пространства посредством Анализа независимых компонент абстракция данных в результате кластеризации подобна абстракции данных исходного потока векторов. Поэтому Анализ независимых компонент может использоваться при абстракции данных для обнаружения паттернов в потоке векторов.

Ключевые слова: *потоки векторов, паттерны, кластеризация, Анализ главных компонент, Анализ независимых компонент*

Введение

В работе [1] рассматривается задача обнаружения паттернов потенциально бесконечных многомерных временных рядов (потоков векторов). Кластеризация временных сечений многомерного временного ряда обеспечивает абстракцию его данных в виде последовательности кодирующих символов — меток кластеров. В этой последовательности символов паттерны многомерного временного ряда представляются повторяющимися эпизодами. Для их обнаружения используется алгоритм подсчета не перекрывающихся эпизодов с ограничением продолжительности. Проекции временных интервалов для повторяющихся эпизодов на исходный многомерный временной ряд обеспечивают обнаружение паттернов этого многомерного временного ряда.

Абстракция данных осуществляется кластеризацией временных сечений $\{x[\overline{1, n}; i], i = \overline{1, N_0}\}$ многомерного временного ряда $\mathbf{X} = x[\overline{1, n}; \overline{1, N_0}]$ (векторов потока) на заданное число кластеров посредством алгоритма K средних [2]. Считается, что набор векторов $\{\mathbf{x}[i], i = \overline{1, N_0}\}$ принадлежит пространству $\mathbb{R}^n$ с Евклидовой мерой расстояния. Кластер объединяет группу векторов (точек в $\mathbb{R}^n$), расстояние между которыми меньше, чем расстояние до векторов за пределами кластера. Сформулируем задачу определения K средних для набора векторов $\{\mathbf{x}[i], i = \overline{1, N_0}\}$ как задачу оптимизации. Пусть векторы $\{\boldsymbol{\mu}[k], k = \overline{1, K}\}$ — это центры кластеров. Введем набор бинарных индикаторов (responsibilities) $\{r[i; k] \in \{0, 1\}, k = \overline{1, K}, i = \overline{1, N_0}\}$, показывающих, какому из K кластеров назначены точки $\{\mathbf{x}[i], i = \overline{1, N_0}\}$ (если точка $\mathbf{x}[i]$ назначена кластеру k, то $r[i; k] = 1$, и $r[i; k] = 0$ для $j \neq k$). Задача оптимизации состоит в назначении точек кластерам с центрами $\{\boldsymbol{\mu}[k], k = \overline{1, K}\}$, при котором целевая функция

$$J = \sum_{i=1}^{N_0} \sum_{k=1}^{K} r[i; k] \|\mathbf{x}[i] - \boldsymbol{\mu}[k]\|^2 \qquad (1)$$

имеет глобальный минимум. После инициализации $\{\boldsymbol{\mu}[k], k = \overline{1, K}\}$ случайными K векторами из $\{\mathbf{x}[i], i = \overline{1, N_0}\}$, минимизация функции J, определяемой по формуле (1) выполняется итеративно, повторяя два шага до достижения условия сходимости. На первом шаге J минимизируется относительно $\{r[i; k], i = \overline{1, N_0}, k = \overline{1, K}\}$ при фиксированных $\{\boldsymbol{\mu}[[k], k = \overline{1, K}\}$. На втором шаге J минимизируется относительно $\{\boldsymbol{\mu}[k], k = \overline{1, K}\}$ при фиксированных $\{r[i; k], i = \overline{1, N_0}, k = \overline{1, K}\}$. Общий подход к кластеризации временных сечений многомерного временного ряда основан на представлении $\{\mathbf{x}[i], i = \overline{1, N_0}\}$ как выборки смеси Гауссовых распределений [3]. Он использует итеративный двухшаговый алгоритм ожидания и максимизации правдоподобия для

кластеризации с индикаторными переменными $\{r[i; k] \in [0, 1], i = \overline{1, N_0}, k = \overline{1, K}\}$ [4]. Однако для инициализации алгоритма, основанного на модели смеси Гауссовых распределений, используется алгоритм K средних. Таким образом, устойчивость алгоритма K средних относительно инициализации центров кластеров $\{\boldsymbol{\mu}[[k], k = \overline{1, K}\}$ определяет устойчивость обоих методов кластеризации.

В качестве примера на рис. 1 показана абстракция данных посредством кластеризации K средних для трехмерного сигнала дистанционного манипулятора на основе MEMS-акселерометра [5]. На рис. 1, *а* координата x сигнала дистанционного манипулятора представлена сплошной линией "—", координата y — штриховой линией "--", координата z — штрих-пунктирной линией "-·". На рис. 1, *б* показана кластеризация трехмерных векторов (точек в $\mathbb{R}^3$), представляющих временные сечения трехмерного ряда на рис. 1, *а*, на шесть кластеров с использованием алгоритма K средних: "+" обозначает точки первого кластера, представляющие временные сечения для интервалов покоя, т. е. пауз между жестами, "×" — второго кластера, "Δ" — третьего кластера, "□" — четвертого кластера, "пятиугольник" — пятого кластера, "шестиугольник" — шестого кластера. Эта кластеризация дает классификацию временных сечений трехмерного ряда на рис. 1, *а*. На рис. 1, *в* представлена абстракция трехмерного временного ряда (рис. 1, *а*): временные сечения трехмерного ряда кодируются как последовательность меток кластеров, т. е. цифр из набора {1, 2, 3, 4, 5, 6}.

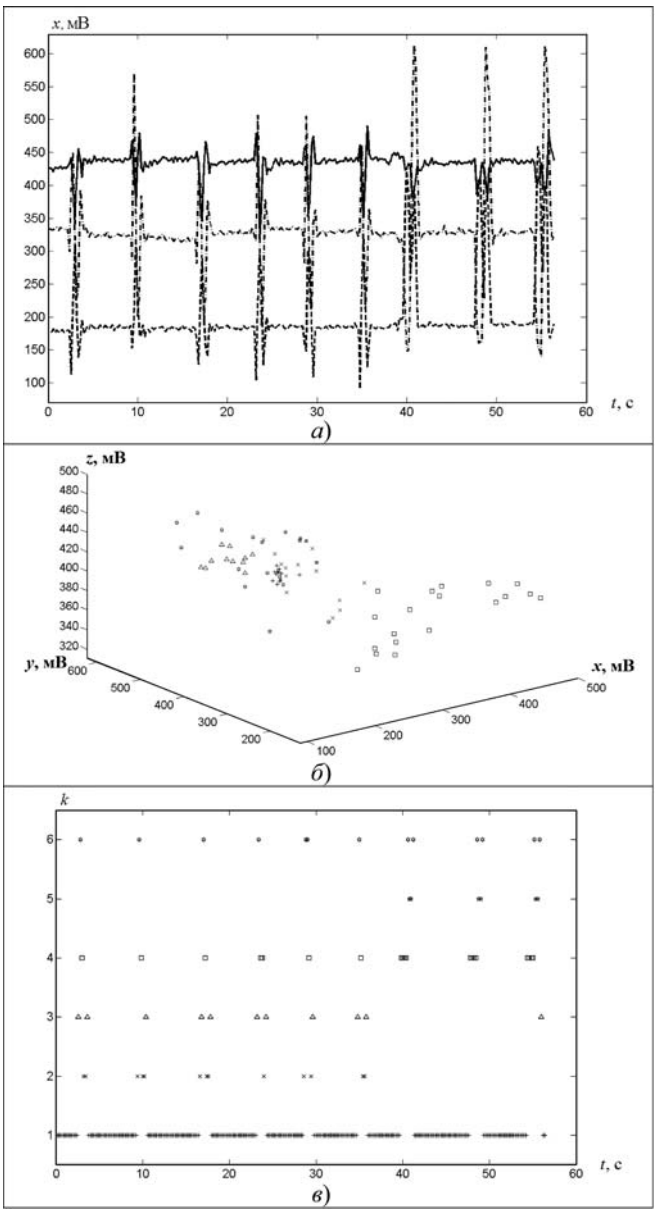


Рис. 1. Абстракция данных для трехмерного сигнала дистанционного манипулятора

Устойчивость абстракции данных на рис. 1, *в* обеспечивается тем, что временные сечения из интервалов покоя (пауз между жестами) доминируют в трехмерном сигнале дистанционного манипулятора (рис. 1, *а*), так как 78 % временных сечений трехмерного ряда на рис. 1, *а* принадлежат интервалам покоя. Поэтому при случайном выборе временного сечения ряда (используемом в алгоритме K средних), как исходного центра первого кластера, с вероятностью 78 % будет выбрано временное сечение, принадлежащее одному из интервалов покоя. В результате при кластеризации посредством алгоритма K средних (K-means) все временные сечения интервалов покоя попадают в первый кластер (рис. 1, *б*). Однако алгоритм K средних гарантирует достижение только локального минимума целевой функции J, определяемой формулой (1). Для достижения глобального минимума J предложен алгоритм K-means++, который использует собственную процедуру инициализации центров кластеров $\{\boldsymbol{\mu}[[k], k = \overline{1, K}\}$ [6, 7].

Пусть $D(\mathbf{x})$ обозначает кратчайшее расстояние от вектора $\mathbf{x}$ до ближайшего центра $\boldsymbol{\mu}[k]$, который уже инициализирован. Алгоритм K-means++ использует следующую инициализацию:

1. Исходный центр $\boldsymbol{\mu}[1]$ выбирается случайно из $\{\mathbf{x}[i], i = \overline{1, N_0}\}$.

2. Для всех $\{\mathbf{x}[i], i = \overline{1, N_0}\}$ вычисляются $\{D(\mathbf{x}[i]), i = \overline{1, N_0}\}$.

3. Случайно генерируется действительное число L, удовлетворяющее условию $0 < L \leqslant \sum_{i=1}^{N_0} D(\mathbf{x}[i])$.

4. В качестве следующего центра $\boldsymbol{\mu}[k]$ выбирается вектор $\mathbf{x}[j]$, удовлетворяющий условию $\sum_{i=1}^{j-1} D(\mathbf{x}[i]) < L \leqslant \sum_{i=1}^{j} D(\mathbf{x}[i])$.

5. Шаги 2—4 повторяются, пока не будут инициализированы все $\{\boldsymbol{\mu}[[k], k = \overline{1, K}\}$.

После инициализации центров кластеров выполняются итерации алгоритма K средних (K-means) до достижения условия сходимости. Инициализация, используемая в алгоритме K-means++, повышает устойчивость результата, приведенного на рис. 1, *в*. Однако в работах [8, 9] показано, что решение (1), определяемое бинарными индикаторами кластеров $\{r[i; k] \in \{0, 1\}, k = \overline{1, K}, i = \overline{1, N_0}\}$, дается *Анализом главных компонент* (Principal Component Analysis (PCA)), и PCA-подпространство, натянутое на векторы главных направлений, идентично подпространству центров кластеров (cluster centroid subspace), определяемому матрицей расстояний между кластерами $\{C[k], k = \overline{1, K}\}$ (between-class scatter matrix) с элементами

$$d(C[k], C[l]) = \sum_{i \in C[k]} \sum_{j \in C[l]} \|\mathbf{x}[i] - \mathbf{x}[j]\|^2.$$

Таким образом, PCA-анализ автоматически проецирует полное пространство на подпространство, где находится глобальный минимум целевой функции J (1), что облегчает поиск почти оптимального решения. PCA-анализ традиционно используется (как предварительная обработка) для сокращения размерности пространства векторов — временных сечений многомерного временного ряда перед их кластеризацией посредством алгоритма K средних [10].

Анализ главных компонент

Для n-мерного ряда $\mathbf{X} = x[\overline{1, n}; \overline{1, N_0}]$ могут быть определены n главных компонент, но считается, что за большую часть вариаций векторов — сечений n-мерного ряда — отвечают m главных компонент, где $m \ll n$ [10]. В пакетной реализации *Анализ главных компонент* осуществляет разложение сингулярных чисел (SVD) матрицы $\mathbf{X}^{\mathrm{T}} \in \mathbb{R}^{N_0 \times n}$ в виде

$$\mathbf{X}^{\mathrm{T}} = \mathbf{U}\Sigma\mathbf{W}^{\mathrm{T}}, \qquad (2)$$

где $\mathbf{U} \in \mathbb{R}^{N_0 \times d}$ — матрица левых сингулярных векторов и $\mathbf{W} \in \mathbb{R}^{n \times d}$ — матрица правых сингулярных векторов, которые являются унитарными, и $\mathbf{\Sigma} \in \mathbb{R}^{d \times d}$ — диагональная матрица

$$\mathbf{\Sigma} = \mathrm{diag}(\sigma[1], \sigma[2], ..., \sigma[d]), \qquad (3)$$

у которой ненулевые элементы положительные и упорядочены по убыванию $\sigma[1] > \sigma[2] > ... > \sigma[d] > 0$. Если в формуле (3) положить $\sigma[i] = 0$, $i = \overline{m, d}$, то отбрасываются наименьшие вариации. Действительно, левые сингулярные векторы $\mathbf{X}^{\mathrm{T}}$ являются собственными векторами $\mathbf{X}^{\mathrm{T}}\mathbf{X} = \mathbf{U}\Sigma\Sigma\mathbf{U}^{\mathrm{T}}$, правые сингулярные векторы $\mathbf{X}^{\mathrm{T}}$ являются собственными векторами $\mathbf{X}\mathbf{X}^{\mathrm{T}} = \mathbf{W}\Sigma\Sigma\mathbf{W}^{\mathrm{T}}$, а ненулевые сингулярные числа $\mathbf{X}$ являются положительными корнями ненулевых собственных чисел $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ и $\mathbf{X}\mathbf{X}^{\mathrm{T}}$ (оценка ковариационной матрицы). Определяя для $\mathbf{X} = [\mathbf{x}[1], ..., \mathbf{x}[n]]^{\mathrm{T}} \in \mathbb{R}^{n \times N_0}$ главные компоненты $\mathbf{S} = [\mathbf{s}[1], ..., \mathbf{s}[d]]^{\mathrm{T}} \in \mathbb{R}^{d \times N_0}$ как $\mathbf{S} = \mathbf{\Sigma}\mathbf{U}^{\mathrm{T}}$, из SVD-разложения (2) для $\mathbf{X}^{\mathrm{T}} \in \mathbb{R}^{N_0 \times n}$ получаем разложение главных компонент в виде

$$\mathbf{X} = \mathbf{W}\mathbf{S}, \qquad (4)$$

где $\mathbf{W} = [\mathbf{w}[1], ..., \mathbf{w}[k]] \in \mathbb{R}^{n \times k}$, $k < n$. Если в формуле (3) $\sigma[i] = 0$, $i = \overline{m, d}$, то выделяются m главных компонент, содержащих большую часть информации n-мерного временного ряда $\mathbf{X} = x[\overline{1, n}; \overline{1, N_0}]$. При вычислении матрицы (2) затраты на память растут с ростом N_0, оцененная матрица $\mathbf{W}$ все сильнее зависит от прошлых данных и все меньше от новых данных. Ясно, что для реальных векторных потоков невозможно выполнять вычисление (2) при поступлении очередного вектора потока, потому что невозможно хранить все прошлые векторы потока. В принципе, все прошлые векторы $\mathbf{x}[i]$, $i \leqslant N_0$ отображаются в текущем значении векторов главных направлений $\mathbf{w}[1]$, ..., $\mathbf{w}[k]$. Поэтому в режиме реального времени для потока векторов используется *Аккумулирующий анализ главных компонент*, обновляющий оценки векторов главных направлений $\mathbf{w}[1]$, ..., $\mathbf{w}[k]$ при поступлении нового вектора потока [11, 12].

В результате *Анализа главных компонент* для трехмерного ряда на рис. 1, *а* определяется ортогональная матрица векторов главных направлений $\mathbf{W}$ из формулы (2)

$$\mathbf{W} = \begin{bmatrix} 0{,}3984 & 0{,}7183 & 0{,}5704 \\ 0{,}5845 & -0{,}6781 & 0{,}4456 \\ 0{,}7069 & 0{,}1558 & -0{,}6900 \end{bmatrix},$$

и по формуле, обратной (4),

$$\mathbf{S} = \mathbf{W}^{\mathrm{T}}\mathbf{X},$$

определяются главные компоненты. На рис. 2, *а* представлены три главные компоненты для трехмерного ряда на рис. 1, *а*. На рис. 2, *а* первая главная компонента представлена сплошной линией "—", вторая главная компонента штриховой линией "--", третья главная компонента — штрих-пунктирной линией "-·". На рис. 2, *б* показана кластеризация временных сечений трех главных компонент посредством алгоритма K средних (обозначения те же, что и на рис. 1, *б*). Рис. 2, *в* показывает абстракцию данных для трех главных компонент. Абстракция данных на рис. 2, *в* отличается от абстракции данных на рис. 1, *в*, но позволяет обнаружить два типа приблизительно повторяющихся эпизодов, которые соответствуют двум паттернам на рис. 1, *а* [1]. Из трех главных компонент на рис. 2, *а* две первые

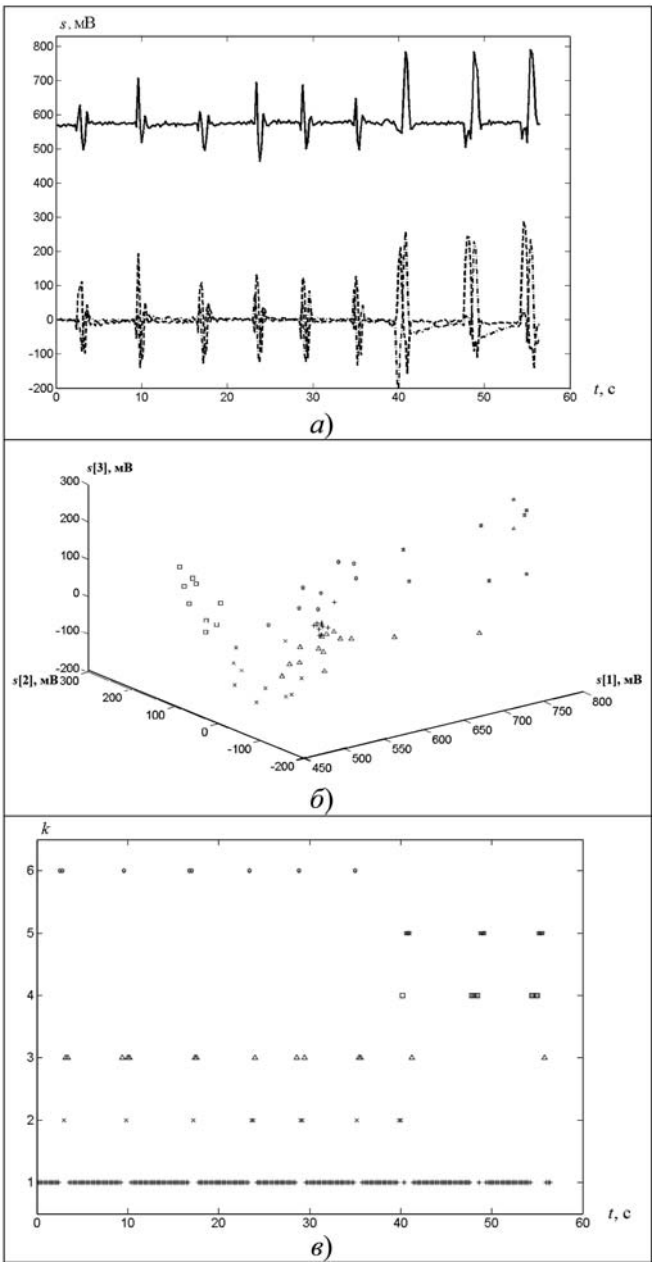


Рис. 2. Абстракция данных для трех главных компонент

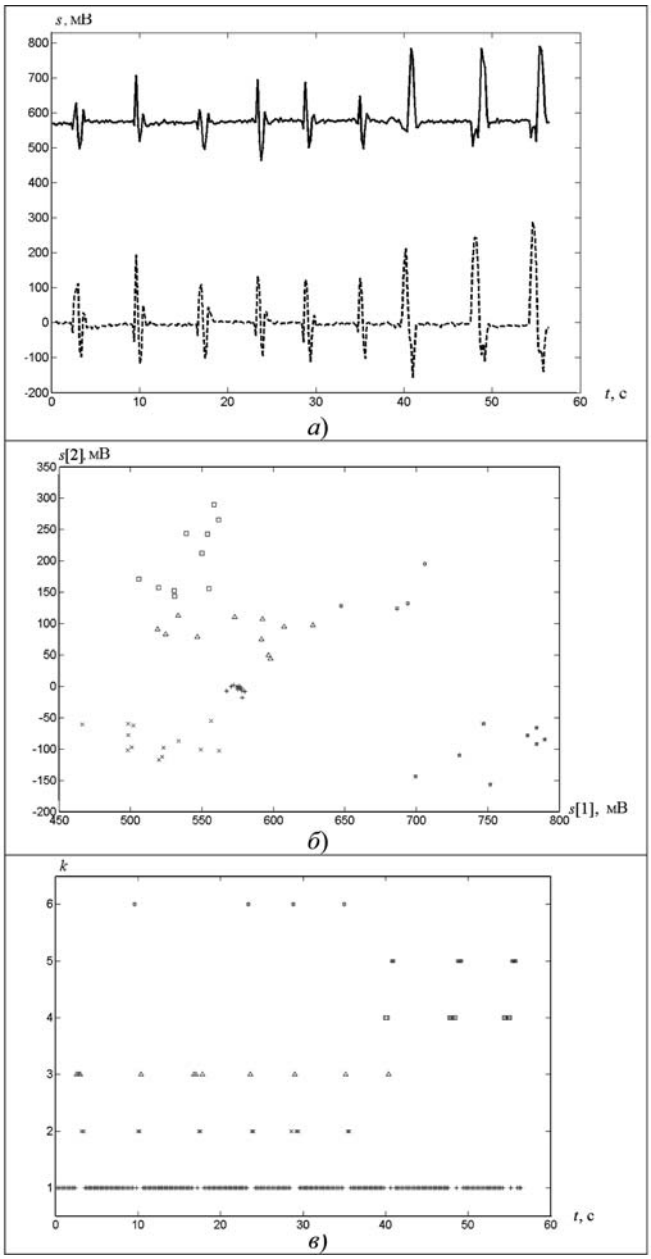


Рис. 3. Абстракция данных для двух главных компонент

являются доминантными. Рис. 3, *б* показывает кластеризацию временных сечений двух доминантных компонент (рис. 3, *а*) посредством алгоритма *K* средних (обозначения те же, что и на рис. 1, *б*). Рис. 3, *в* показывает абстракцию данных. Потеря энергии при переходе к двум доминантным компонентам (рис. 3, *а*) приводит к тому, что последовательности эпизодов на рис. 2, *в* воспроизводятся на рис. 3, *в* только приближенно (т. е. они дефектные). Поэтому в следующем разделе для сокращения размерности многомерного временного ряда перед кластеризацией посредством алгоритма *K* средних используется *Анализ независимых компонент* (ICA) [13].

Анализ независимых компонент

В работах [14, 15] *Анализ независимых компонент* применяется для инициализации центров кластеров для алгоритма *K* средних, гарантирующей определение глобального минимума целевой функции *J*. Однако для рассматриваемой задачи абстракции данных многомерных временных рядов этот подход не может быть применен. Он предполагает, что число кластеров мало по сравнению с размерностью данных временного ряда, а в задаче абстракции данных многомерного временного ряда всегда выполняется обратное соотношение. В настоящем разделе на примере записей сигналов дистанционного манипулятора показано, что *Анализ независи-*

мых компонент способен выделять доминантные компоненты многомерных временных рядов [16, 17].

Разложение независимых компонент для многомерного временного ряда $\mathbf{X} = x[\overline{1, n}\ ;\ \overline{1, N_0}]$ имеет вид

$$\mathbf{X} = \mathbf{AS}, \tag{5}$$

где $\mathbf{A} \in \mathbb{R}^{n \times m}$ — постоянная матрица смешивания, а строки матрицы $\mathbf{S} \in \mathbb{R}^{m \times N_0}$ — это выборки для статистически независимых случайных переменных с не Гауссовыми распределениями и средним значением, равным нулю. Соответственно, строки $\mathbf{X}$ также считаются выборками случайных переменных со средним значением, равным нулю. Обратная формула имеет вид

$$\mathbf{S} = \mathbf{BX}, \tag{6}$$

где $\mathbf{B}$ — матрица весов. Таким образом, *Анализ независимых компонент* — это не Гауссова версия факторного анализа [13].

В рамках *Анализа независимых компонент* (ICA) разработано большое число алгоритмов оценки матрицы весов $\mathbf{B}$ [13]. Однако для импульсных многомерных временных рядов типа записей трехмерного сигнала дистанционного манипулятора на рис. 1, *а* эти алгоритмы дают подобные независимые компоненты с точностью до перестановки и масштабирования. В этом разделе для *Анализа независимых компонент* многомерных временных рядов используется предложенный в работе [18] алгоритм SOBI (Second-Order Blind Identification) и предложенный в работе [19] алгоритм JADE (Joint Approximation Diagonalization of Eigen-matrices).

Алгоритм SOBI описан в статье [16]. Определим строки $\mathbf{X}[i] = x[i;\ \overline{1, N_0}]$, $i = \overline{1, n}$, матрицы данных $\mathbf{X}$. В алгоритме SOBI исходные данные $\mathbf{X}$ с нулевыми средними строк $\mathbf{X}[i]$, $i = \overline{1, n}$, отбеливаются с помощью матрицы $\mathbf{U}$ из формулы (2), которая диагонализует оценку ковариационной матрицы $\mathbf{XX}^{\mathrm{T}}$. Для отбеленных данных $\mathbf{X}$ строятся ковариационные матрицы с запаздыванием $\tau = 1, 2, 3, 4, \ldots$, и они совместно приближенно диагонализуются посредством определяемой ортогональной матрицы $\mathbf{V}$, которая превращает отбеленные данные $\mathbf{X}$ в максимально статистически независимые. Оценка для матрицы смешивания $\mathbf{A}$ из (5) имеет вид

$$\mathbf{A} = \mathbf{V}^{\mathrm{T}}\mathbf{U}. \tag{7}$$

Если $\mathbf{A}$ (7) является матрицей полного ранга по столбцам (full column-rank), то независимые компоненты $\mathbf{S}$ определяются из (5) по методу наименьших квадратов и в (6) матрица весов $\mathbf{B} = \mathbf{A}^{+}$ ("+" — это операция псевдообращения матриц) [16].

В то время как алгоритм SOBI основан на моментах (статистиках) второго порядка, алгоритм JADE использует статистики четвертого порядка. В алгоритме JADE считается, что исходные данные $\mathbf{X}$ с нулевыми средними строк $\mathbf{X}[i]$, $i = \overline{1, n}$, и отбелены с помощью матрицы $\mathbf{U}$ из (2). Считается также, что распределения случайных переменных $\mathbf{X}[i]$, $i = \overline{1, n}$, симметричные, т. е. все нечетные моменты равны нулю, и рассматриваются только моменты (cumulants) четвертого порядка

$$Q^{\mathbf{x}}[i; j; k; l] = \mathrm{E}[\mathbf{X}[i], \mathbf{X}[j], \mathbf{X}[k], \mathbf{X}[l]]. \tag{8}$$

Алгебраическая природа моментов (cumulants) (8) тензорная, но на основе этих статистик четвертого порядка $Q[i; j; k; l]$ в работе [19] строятся специальные матрицы моментов. Для заданных случайных переменных $\mathbf{X}[i] \in \mathbb{R}^{1 \times N_0}$, $i = \overline{1, n}$, и произвольной матрицы $\mathbf{M} \in \mathbb{R}^{n \times n}$ ассоциированная матрица моментов (cumulant matrix) $Q^{\mathbf{x}}(\mathbf{M})$ имеет элементы

$$Q^{\mathbf{x}}(\mathbf{M})[i; j] = \sum_{k, l} Q^{\mathbf{x}}[i; j; k; l]M[k; l]. \tag{9}$$

В работе [19] показано, что матрица смешивания $\mathbf{A}$ в (5) может быть оценена по (7), где $\mathbf{V}$ — ортогональная матрица, которая совместно приводит все матрицы моментов (cumulant matrices) $Q^{\mathbf{x}}(\mathbf{M})$ (9) к приблизительно диагональному виду в соответствии с функцией контраста JADE (JADE contrast function), являющейся эффективной мерой перекрестной корреляции независимых компонент $\mathbf{S}$ из (5):

$$\varphi^{\mathrm{JADE}}(\mathbf{S}) = \sum_{i, j, k, l \neq i, i, k, k} (Q^{\mathbf{s}}[i; j; k; l])^2. \tag{10}$$

Алгоритмы SOBI и JADE усовершенствованы в работе [20] с помощью PARAFAC-разложения тензоров третьего порядка [12, 21]. При PARAFAC-разложении не требуется, чтобы матрица $\mathbf{A}$ (7) была длинной и полного ранга по столбцам [20].

В отличие от *Анализа главных компонент*, в котором число компонент ограничивается в (3), анализ независимых компонент проводится с конкретным числом компонент. В результате анализа для двух независимых компонент трехмерного ряда на рис. 1, *а* определяется матрица весов $\mathbf{B}$ из (6):

$$\mathbf{B} = \begin{bmatrix} 0{,}0107 & 0{,}0120 & -0{,}0032 \\ 0{,}0141 & -0{,}0093 & -0{,}0033 \end{bmatrix}.$$

На рис. 4, *а* представлены определенные по формуле (6) две независимые компоненты для трехмерного ряда на рис. 1, *а*. На рис. 4, *а* первая независимая компонента представлена сплошной линией "—", вторая независимая компонента штриховой линией "--". Рис. 4, *б* показывает кластеризацию временных сечений двух независимых компонент посредством алгоритма *K* средних (обозначения те же, что и на рис. 1, *б*). Рис. 4, *в* показывает абстракцию данных для двух независимых компонент.

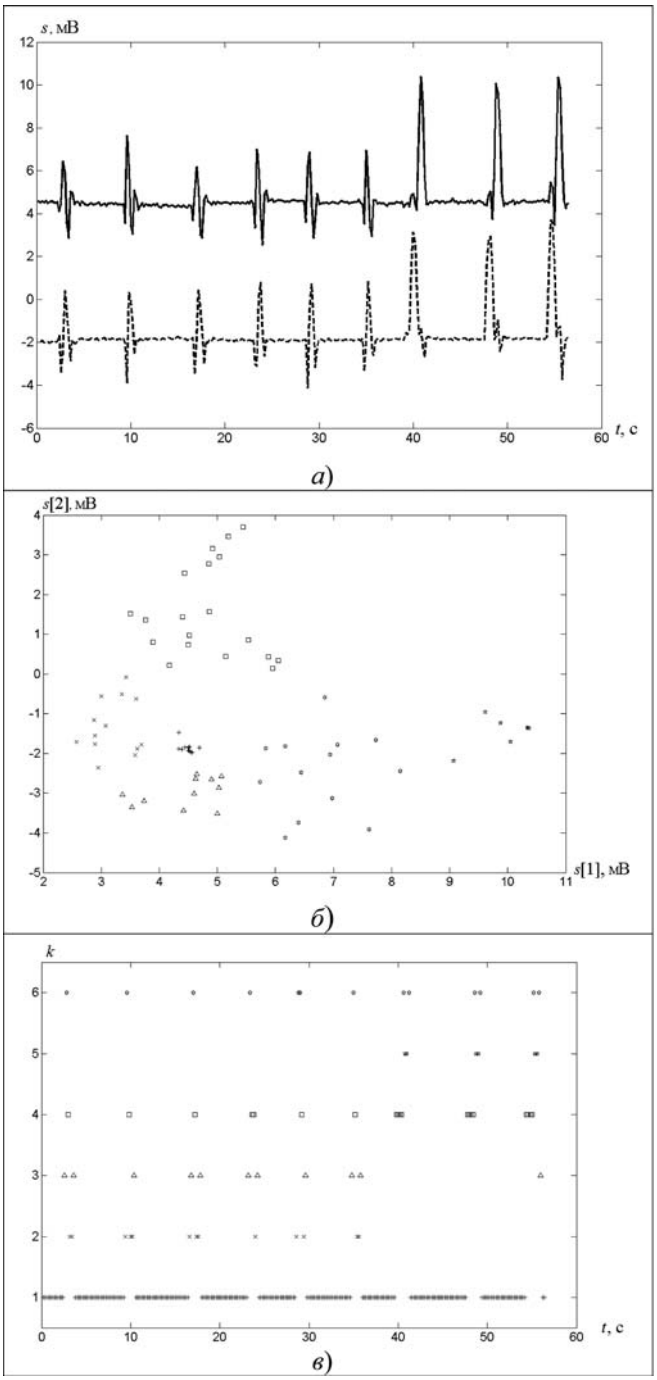


Рис. 4. Абстракция данных для двух независимых компонент

Абстракция данных для двух независимых компонент на рис. 4, *в* подобна абстракции данных на рис. 1, *в* для трехмерного сигнала дистанционного манипулятора. В принципе, присутствие паттернов, представляющих два типа жестов оператора в трехмерном сигнале дистанционного манипулятора на рис. 1, *а*, проявляется непосредственно в представлении независимых компонент этого сигнала на рис. 4, *а*. Поэтому *Анализ независимых компонент* можно использовать при абстракции данных для обнаружения паттернов в потенциально бесконечном многомерном временном ряде — потоке векторов.

Заключение

Для обеспечения устойчивости алгоритма кластеризации многомерных векторов на заданное число кластеров (*K*-means) традиционно используется предварительная обработка набора векторов на основе *Анализа главных компонент*, проецирующего вектора в подпространство сокращенной размерности. Показано, что при использовании алгоритма кластеризации на заданное число кластеров с целями построения абстракции данных для потока векторов (временных сечений многомерного временного ряда), абстракция данных, как результат кластеризации временных сечений главных компонент потока векторов, только приближенно представляет абстракцию данных исходного потока векторов. Однако при сокращении размерности пространства векторов посредством *Анализа независимых компонент* абстракция данных, как результат кластеризации временных сечений независимых компонент, подобна абстракции данных исходного потока векторов. Поэтому *Анализ независимых компонент* может использоваться как (сокращающая размерность векторов) предварительная обработка при абстракции данных для обнаружения паттернов в потоке векторов.

Список литературы

1. **Кухаренко Б. Г., Пономарев Д. И.** Обнаружение паттернов многомерных временных рядов на основе абстракции данных // Информационные технологии. 2013. № 4. С. 28—34.
2. **Han J., Kamber M.** Data Mining: Concepts and Techniques. 2nd ed. New York: Morgan Kaufmann Publishers. 2006.
3. **Bouman C. A.** CLUSTER: An Unsupervised Algorithm for Modeling Gaussian Mixtures. Purdue University, West Lafayette: School of Electrical Engineering, 2005. P. 1—20.
4. **Dempster A., Laird N. M., Rubin D. B.** Maximum likelihood from incomplete data via the EM algorithm // Journal of the Royal Statistical Society B. 1977. V. 39. N 1. P. 1—38.
5. **Кухаренко Б. Г., Пономарев Д. И.** Дистанционный манипулятор на основе MEMS-акселерометра в качестве чувствительного элемента // Нано- и микросистемная техника. 2012. № 2. С. 49—54.
6. **Arthur D., Vassilvitskii S.** *K*-means++: the advantages of careful seeding // SODA'07 Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms. Philadelphia, PA: SIAM Press. 2007. P. 1027—1035.
7. **Bahmani B., Moseley B., Vattani A., Kumar R., Vassilvitskii S.** Scalable *K*-means++ // Proceedings of VLDB Endowment. 2012. V. 5. N 7. P. 622—633.
8. **Zha H., Ding C., Gu M., He X., Simon H. D.** Spectral relaxation for *K*-means clustering // Neural Information Processing Systems (NIPS 2001). Cambridge, MA: MIT Press. 2001. V. 14. P. 1057—1064.
9. **Ding C., He X.** *K*-means clustering via Principal Component Analysis // Proceedings of the Twenty-first International Conference on Machine Learning (ICML 2004). Banff, Alberta, Canada: ACM Press. 2004. P. 225—232.
10. **Jolliffe I. T.** Principal Component Analysis. New York, Berlin, Heidelberg: Springer. 2002.
11. **Papadimitriou S., Sun J., Faloutsos C.** Streaming pattern discovery in multiple time-series // Proceedings of the 31st Very Large Data Bases Conference (VLDB'05). Trondheim, Norway: VLDB Endowment. 2005. P. 697—708.
12. **Кухаренко Б. Г.** Аккумулирующий анализ главных компонент потоков тензорных данных // Информационные технологии. 2013. № 2. Приложение. С. 1—32.

13. **Cichocki A., Amari S.** Adaptive Blind Signal and Image Processing: Learning Algorithms and Applications. Chichester, West Sussex, UK: John Wiley & Sons. 2002.

14. **Onoda T., Sakai M., Yamada S.** Careful seeding method based on Independent Components Analysis for *K*-means clustering // 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology. Toronto, Canada: IEEE Computer Society. 2010. P. 112—115.

15. **Onoda T., Sakai M., Yamada S.** Careful seeding method based on Independent Components Analysis for *k*-means clustering // Journal of Emerging Technologies in Web Intelligence. 2012. V. 4. N 1. P. 51—59.

16. **Кухаренко Б. Г.** Анализ независимых компонент записей колебаний в технологии спектрального анализа на основе быстрого преобразования Прони // Информационные технологии. 2009. № 9. С. 20—25.

17. **Кухаренко Б. Г.** Анализ независимых компонент и скрытая Марковская модель для определения доминантных компонент многомерных временных рядов // Информационные технологии. 2010. № 11. Приложение. С. 1—32.

18. **Belouchrani A., Abed-Meraim K., Cardoso J.-F., Molines E.** A blind source separation technique using second-order statistics // IEEE Transactions on Signal Processing. 1997. V. 45. P. 434—444.

19. **Cardoso J.-F.** High-order contrasts for independent component analysis // Neural Computation. 1999. V. 11. Issue 1. P. 157—192.

20. **Nion D., Sidiropoulos N. D.** Adaptive algorithms to track the PARAFAC decomposition of a third-order tensor // IEEE Transactions on Signal Processing. 2009. V. 57. N 6. P. 2299—2310.

21. **Kolda T. G., Bader B. W.** Tensor Decompositions and Applications // SIAM Review. 2009. V. 51. N 3. P. 455—500.

УДК 519.1

Р. А. Сологуб, аспирант,
e-mail: roman.sologub@yahoo.com,
Вычислительный Центр РАН
им. А. А. Дородницына, г. Москва

Алгоритмы порождения нелинейных регрессионных моделей[1]

Рассматривается задача порождения нелинейных моделей при решении задач восстановления регрессии. Регрессионной моделью называется параметрическое семейство функций. Каждая из порождаемых моделей является суперпозицией функций из некоторого экспертно заданного множества. Эти функции называются примитивами, или порождающими функциями. Для создания модели выбирается набор порождающих функций. Оценка параметров модели проводится с помощью метода Левенберга—Марквадта. Выбор моделей выполняется с помощью скользящего контроля. Далее на каждом шаге с помощью методов символьной регрессии, генетического программирования и трансформации графов показывается возможность модификации и уточнения исходных моделей.

Ключевые слова: *символьная регрессия, теория графов, порождение моделей нелинейной регрессии, преобразование графов*

Введение

Индуктивное порождение моделей с помощью методов группового учета аргумента рассматривается в работах Г. Н. Ивахненко [1, 6]. Алгоритм целенаправленно перебирает модели-претенденты различной сложности по ряду критериев. В результате находится модель оптимальной структуры.

Использование нелинейной регрессии для решения прикладных задач описывается в работах Дж. Себера [2, 3]. В них предлагается построение и оценка параметров нелинейных моделей. Для оценки параметров моделей используется алгоритм Левенберга—Марквадта.

Использование нелинейной регрессии для индуктивного порождения моделей с помощью эволюционных алгоритмов рассматривается в работах Дж. Козы и И. Зелинки [4, 7]. Для построения моделей применяется метод символьной регрессии. Главная цель эволюционных алгоритмов состоит в том, чтобы при случайном поиске найти во время эволюционного процесса лучшее решение. Символьная регрессия основана на эволюционных алгоритмах, и ее главная цель в контексте работы — породить эволюционным способом такую модель, которая приблизит данные наилучшим возможным образом. В этом методе модели, являющиеся суперпозициями гладких функций, представляются в виде графов-деревьев. Для создания моделей Д. Коза [4] предлагает использовать методы генетического программирования: генетическое скрещивание моделей и генетическую мутацию. При этом авторы ставят ряд нерешенных вопросов: появление моделей с ненастраиваемыми параметрами, деление на ноль, возникновение комплексных аргументов. Часть возникающих вопросов разбирается в предыдущей работе автора [14] и в данной работе.

Для упрощения структуры моделей используются методы трансформации графов, предложенные в работах Мори [13]. Для преобразования деревьев выделяются некоторые элементарные графы, для которых строятся оболочки изоморфных им графов более сложной структуры.

Для недопущения эффекта переобученности структурная сложность моделей должна ограничиваться. Сложность моделей в рамках данной работы оценивается по методу, предложенному К. Владиславлевой [11]. Для модели, представленной в виде

[1] Работа поддержана РФФИ, проект 10-07-00422.

дерева, ее сложность равна числу элементов во всех поддеревьях данного дерева.

В качестве иллюстрации предложенного подхода рассматривается задача определения компаний-банкротов по набору экспертных оценок и финансовых показателей, выраженных в виде бинарных признаков.

Постановка задачи

Пусть задана выборка $D = \{\{\mathbf{x}_n, y_n\}\}_{n=1}^{N}$, $\mathbf{x} \in R^m$. Выборка разбивается на обучающую и тестовую. Обозначим l, C — множества индексов из $\{1, ..., N\} = W$. Эти множества удовлетворяют условиям разбиения $l \cup C = W$, $l \cap C = \varnothing$. Матрица $\mathbf{X}_l$ состоит из тех векторов-строк $\mathbf{x}_n$, для которых индекс $n \in l$. Вектор $\mathbf{y}_l$ состоит из тех элементов y_n, для которых индекс $n \in l$. Разбиение выборки представляется в виде

$$X_W = \begin{pmatrix} X_l \\ X_C \end{pmatrix}; \mathbf{y}_W = \begin{pmatrix} y_l \\ y_C \end{pmatrix},$$

где $\mathbf{y}_W \in R^{N \times 1}$, $X_W \in R^{N \times m}$, $|l| + |C| = N$.

Требуется построить отображение $\varphi(\mathbf{x}, \mathbf{w}) \to R^1$. Требуется определить модель f — отображение из декартова произведения множества свободных переменных $\mathbf{x} \in R^n$ и множества параметров $\mathbf{w} \in R^m$ в R^1. Модель должна соответствовать отображению φ. Для модели оценивается набор параметров $\mathbf{w}_0$, доставляющий минимум функции квадратичной ошибки

$$S(\mathbf{w}|D, f) = \sum_{n=1}^{N} (f(\mathbf{x}_n, \mathbf{w}) - y_k)^2.$$

Выражение $S(\mathbf{w} \mid D, f)$ означает значение функции ошибки S, которое зависит от набора параметров $\mathbf{w}$ при заданной выборке D и модели f. Такая модель будет называться оптимальной при условии, что ее сложность $C(f)$ не превышает заданной. Сложность определяется как число элементов во всех поддеревьях, которые можно выделить из дерева, представляющего модель.

Задано множество G порождающих функций $g(\mathbf{w}, \mathbf{x})$. Для каждого элемента данного множества g_i определены области аргументов $\mathbf{w} \in R^m$, $\mathbf{x} \in R^n$ и значений, при этом область значений принадлежит R^1. В множество порождающих функций обязательно входит не имеющая аргументов функция $id(\mathbf{x})$, значение которой тождественно значению свободной переменной. Порождается множество моделей $f \in F$ — допустимых суперпозиций, состоящих из функций $g_i \in G$. Требуется выбрать модель, доставляющую минимум $S(f \mid \mathbf{w}^*, D)$ при условии, накладываемом на сложность $C(f) < C^*$.

Алгоритмы порождения моделей

В данном разделе будут рассмотрены различные методы порождения и модификации моделей, основанные на идее построения моделей как суперпозиций порождающих функций.

Построение всех возможных линейных моделей. Для случая, когда множество G содержит только элементы $\{+, id, x\}$, описание и перебор всех возможных моделей наилучшим образом задается с помощью вектора $\mathbf{a} = [a_1, ..., a_n]$, где n — общее число свободных переменных, а $a_i \in \{0, 1\}$ в зависимости от того, включается данная переменная в модель $f = \sum_{i=1}^{N} a_i x_i$ или нет.

Таким образом, число возможных моделей для данного случая равно 2^n.

Построение всех возможных нелинейных моделей. Простейшим способом выбора лучшей модели является полный перебор множества моделей. Модели порождаются рекурсивно (для каждого нового элемента суперпозиции v_i подставляются все возможные для него порождающие функции $g_i \in G$), при этом порождение идет от функции $id(\mathbf{x})$. Процесс конечен, так как ограничивается сложность порождаемых моделей. Ниже приведена процедура построения всех возможных моделей.

```
ПРОЦЕДУРА ПостроениеМодели(ТекущаяМодель,
Примитивы, Родитель)
для j = 1, ..., size(Примитивы)
   НовыйЭлемент := Примитивы(j);
   ВременнаяМодель := [ТекущаяМодель; Новый Эле-
   мент];
   если ПроверкаСложности(ВременнаяМодель) == 1 то
      ТекущаяМодель := ВременнаяМодель;
      НовыеЭлементы = Примитивы (НовыйЭле-
      мент).ЧислоДетей;
      для i = 1, ..., НовыеЭлементы
         ТекущаяМодель := ПостроениеМодели(Теку-
         щаяМодель, Примитивы, НовыйЭлемент);
      return ТекущаяМодель
выход
```

Метод группового учета аргументов (МГУА). МГУА — метод порождения и выбора обобщенно-линейных регрессионных моделей оптимальной сложности (функциями-примитивами таких моделей могут быть только умножение и сложение). Под сложностью модели в МГУА понимается число параметров. Для порождения используется базовая модель, подмножество элементов которой должно входить в искомую модель.

Индуктивный алгоритм отыскания модели оптимальной структуры состоит из следующих основных шагов. Назначается базовая модель. Эта модель описывает отношение между зависимой переменной y и свободными переменными $\mathbf{x}$. Например, используется функциональный ряд Воль-

терра, называемый также полиномом Колмогорова—Габора:

$$y = w_0 + \sum_{i=1}^{m} w_i x_i + \sum_{i=1}^{m}\sum_{j=1}^{m} w_{ij} x_i x_j + \sum_{i=1}^{m}\sum_{j=1}^{m}\sum_{k=1}^{m} w_{ijk} x_i x_j x_k + \ldots .$$

В этой модели $\mathbf{x} = \{x_i | i = 1, \ldots, m\}$ — множество свободных переменных и $\mathbf{w}$ — вектор параметров — весовых коэффициентов

$$\mathbf{w} = \langle w_i, w_{ij}, w_{ijk}, \ldots | i, j, k, \ldots = 1, \ldots, m \rangle.$$

Базовая модель линейна относительно параметров w и нелинейна относительно свободных переменных x_i.

Индуктивно порождаются модели-претенденты. При этом вводится ограничение на длину полинома базовой модели: степень полинома базовой модели не должна превышать заданное число R. Тогда базовая модель представима в виде линейной комбинации заданного числа произведений свободных переменных

$$y = f(x_1, x_2, \ldots, x_1^2, x_1 x_2, x_2^2, \ldots, x_m^R),$$

здесь f — линейная комбинация. Аргументы этой функции переобозначаются следующим образом:

$$x_1 \mapsto a_1, x_2 \mapsto a_2, \ldots, x_1^2 \mapsto a_\alpha,$$

$$x_1 x_2 \mapsto a_\beta, x_2^2 \mapsto a_\gamma, \ldots, x_m^q \mapsto a_{F_0},$$

т. е.

$$y = f(a_1, a_2, \ldots, a_{F_0}).$$

Для линейно входящих коэффициентов задается одноиндексная нумерация $\mathbf{w} = \langle w_1, \ldots, w_{F_0} \rangle$. Тогда модель может быть представлена в виде линейной комбинации

$$y = w_0 + \sum_{i=1}^{F_0} w_i a_i = w_0 + \mathbf{w} \cdot \mathbf{a}.$$

Каждая порождаемая модель задается линейной комбинацией элементов $\{(w_i, a_i)\}$, в которой множество индексов $\{i\} = s$ является подмножеством $\{1, \ldots, F_0\}$.

Символьная регрессия и генетическое программирование. Для описания процедуры порождения нелинейных моделей сложной структуры необходим язык описания моделей, легко интерпретируемый как пользователем — экспертом в предметной области, так и программной системой. Каждой модели поставим в соответствие направленный граф — дерево $\Gamma = \langle V, E \rangle$. Каждой вершине $v_i \in V$ соответствует порождающая функция g_i. Число ветвей $e_i \in E$, выходящих из каждой вершины v_i, будет равно числу аргументов порождающей функции g_i, соответствующей данной вершине. Листьями дерева Γ являются порождающие функции, не имеющие аргументов, — константы $id(C)$ и свободные переменные $id(\mathbf{x})$.

Выбор оптимальной модели происходит на множестве порождаемых моделей на каждой итерации эволюционного алгоритма. Задан начальный набор конкурирующих моделей $F_0 = \{f_1, \ldots, f_M | f \in \Phi\}$, в котором каждая модель f_i есть суперпозиция порождающих функций $\{g_{ij}\}_{j=1}^{r_i}$.

Алгоритм, который работает итерационно, пока не будет достигнут необходимый уровень значения ошибки или через определенное число итераций, состоит из следующих шагов.

1. Методом сопряженных градиентов (или другим способом настройки параметров) минимизируются функции ошибки $S_i(\mathbf{w})$ для каждой модели f_i, $i = 1, \ldots, M$. Отыскиваются параметры $\mathbf{w}_i^m$ и вычисляется значение функции ошибки каждой модели.

2. Заданы следующие правила построения производных моделей $f_1', \ldots, f_M'$. Для каждой модели f_i строится производная модель f_i'. В f_i произвольно выбирается функция g_{ij}. Выбирается произвольная модель $f_x i$ из $F_0 \backslash \{f_i\}$ и ее произвольная функция $g_{\xi\zeta}$. Модель f' порождается из модели f путем замещения функции g_{ij} с ее аргументами на функцию $g_{\xi\zeta}$ с ее аргументами.

3. С заданной вероятностью η каждая модель f_i' подвергается изменениям. В изменяемой модели выбирается j-я функция, причем закон распределения вероятности выбора функции $p(j)$ задан. Из множества G случайным образом выбирается функция g' и замещает функцию g_j.

4. При выборе моделей из объединенного множества родительских и порожденных моделей в соответствии с критерием $S(\mathbf{w})$ выбираются M наилучших, которые используются в дальнейших итерациях.

Упрощение моделей

Для построения оптимальной модели f ограниченной сложности $C^* < C(f)$ необходимо найти способ трансформации большей модели (представляемой в виде графа Γ с большим числом элементов) в меньшую с помощью специальной процедуры упрощения.

Процедура упрощения минимизирует размер дерева при условии сохранения значений функции, соответствующей данному дереву.

Существуют, по меньшей мере, два широко используемых метода упрощения моделей: алгебраическое упрощение и упрощение эквивалентным решением [13].

Упрощение Соула [12] может быть рассмотрено как алгебраическое упрощение, в котором цели упрощения являются порождающими функциями.

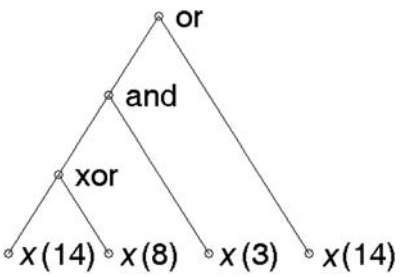


Рис. 1. Модель 1

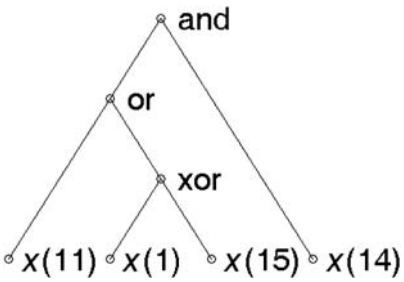


Рис. 2. Модель 2

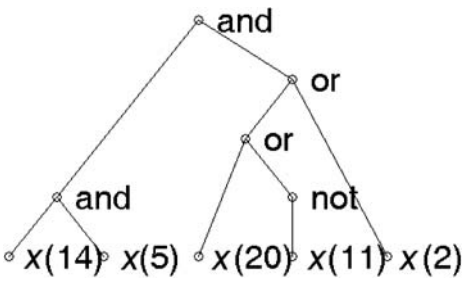


Рис. 3. Модель 3

Тем не менее, существуют примеры, не упрощаемые с помощью подобных техник, например $(X - 1) + (1 - X)$.

Эквивалентное упрощение строится по принципиально другому сценарию. В данном случае равенство моделей определяется не семантически, а численно. В таком случае, два выражения, дающие равные значения на некотором множестве, считаются эквивалентными.

К примеру, при наборе порождающих функций {+, −, *, /} упрощение по сути является приведением к наиболее короткой алгебраической записи. При этом возможно частичное управление с помощью правил типа $1 * X = X$ или $0 * X = 0$. В символьной регрессии эквивалентное упрощение может проведено следующим образом.

1. Выбирается набор простых суперпозиций Θ.

2. Все поддеревья в выбранном дереве проверяются на эквивалентность деревьям в Θ.

3. Если какое-либо поддерево в дереве эквивалентно дереву из Θ, данное поддерево заменяется элементом из Θ.

Процедура повторяется до тех пор, пока после итерации дерево не остается неизменным. Дополнительно вводится алфавитное упорядочение для ветвей, выходящей из вершины, соответствующей коммутативному оператору.

Эквивалентное упрощение расширяет алгебраическое упрощение, позволяя проводить операции:

- слишком сложные в рамках алгебраического подхода;
- верные везде, кроме множеств нулевой меры;
- верные на множестве входов модели. С точки зрения обучаемого алгоритма такие выражения оказываются эквивалентными.

Вычислительный эксперимент

В качестве иллюстративного примера методов порождения моделей, описанных в статье, исследовалась задача определения успешности предприятий по внешним критериям.

Использованы данные о фирмах, кредитовавшихся в определенном турецком банке в 2002—2006 годах. Данные компании работали в сфере производства. Из 627 фирм 61 потерпела дефолт за четырехлетний период. Качественные оценки, данные экспертами и Турецким Центральным Банком сохранены в бинарных шкалах. Числовые значения переведены в бинарные согласно экспертным оценкам. Финансовые отчетности, на основании которых собраны исходные данные, были подготовлены и проверены независимыми аудиторами перед подачей в банк.

Всего используется 21 независимая переменная. Значение входных переменных соответствует экспертной оценке по каждому критерию для каждой фирмы. Значение выходной переменной равно 1, если компания успешно функционировала на всем периоде, и 0, если компания потерпела дефолт на рассматриваемом периоде.

Результаты работы наилучших моделей представлены в таблице.

Номер модели	Сложность	S на обучающей выборке	S на тестовой выборке
1	19	6,23	6,32
2	48	6,09	6,45
3	26	5,73	6,15
Экспертная модель	—	5,56	6,24

Представление наилучших моделей в виде дерева приведено на рис. 1—3.

Заключение

Предложен алгоритм описания, конструктивного порождения и упрощения регрессионных моделей. Приведен способ представления моделей в виде суперпозиций заданных параметрических функций. Описан способ порождения всех моделей заданных классов с помощью алгоритма. Представлена методология упрощения моделей как на основе исследования структуры модели, так и на основе значений модели на входных данных. Для иллюстрации работы алгоритма рассмотрена модель зависимости вероятности банкротства компании от макроэкономических показателей данной компании. Модели, полученные в ходе вычислительного эксперимента, оказываются предпочтительнее моделей, которые были предложены экспертами ранее.

Список литературы

1. **Ивахненко А. Г.** Индуктивный метод самоорганизации моделей сложных систем. Киев: Наукова думка. 1981.
2. **Seber G. A. F., Wild C. J.** Nonlinear Regression. New-York: Wiley—IEEE, 2003.
3. **Seber G. A. F., Schwarz C. J.** Estimating Animal Abundance: Review III. Stat Sci. 1999. Vol. 14, N 4. P. 427—456.
4. **Koza J. R., Keane M. A., Rice J. P.** Performance improvement of machine learning via automatic discovery of facilitating functions as applied to a problem of symbolic system identification // IEEE International Conference on Neural Networks I. San Francisco, USA, 1993. P. 191—198.
5. **Levenberg K.** A Method for the Solution of Certain Non-Linear Problems in Least Squares // The Quarterly of Applied Mathematics. 1944. Vol. 2. P. 164—168.
6. **Malada H. R., Ivakhnenko A. G.** Inductive Learning Algorithms for Complex Systems Modeling. London: CRC Press. 1994.
7. **Zelinka I., Nolle L., Oplatkova Z.** Analytic Programming — Symbiloc Regression by Means of Arbitrfary Evolutionary Algorithms // Journal of Simulation. 2003. Vol. 6. N 9. P. 44—56.
8. **Comisky W., Yu J., Koza J. R.** Automatic synthesis of a wire antenna using genetic programming // Late Breaking Papers at the 2000 Genetic and Evolutionary Computation Conference, Las Vegas, Nevada. 2000. P. 179—186.
9. **Стрижов В. В.** Поиск параметрической регрессионной модели в индуктивно заданном множестве // Журнал вычислительных технологий. 2007. № 1. С. 93—102.
10. **Стрижов В. В., Сологуб Р. А.** Индуктивное построение регрессионных моделей волатильности опционов // Журнал вычислительных технологий. 2009. № 5. С. 102—113.
11. **Vladislavleva E., Smith G., Hertog D.** Order of nonlinearity as a complexity measure for models generated by symbolic regression via pareto genetic programming // IEEE Transactions on Evolutionary Computation. 2006. Vol. 13 (2). P. 333—349.
12. **Soule T., Foster J. A.** Support for multiple causes of code growth in GP. 20. July 1997, working papers of the Workshop on Evolutionary Computation with Variable Size Representation at ICGA-97. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.52.3306 &rep=repl&type=ps
13. **Mori N., McKay B., Hoai N., Essam D., Takeuchi S.** A New Method for Simplifying Algebraic Expressions in Genetic Programming Called Equivalent Decision Simplification // IWANN. 2009. N 2. P. 171—178.
14. **Сологуб Р. А.** Порождение регрессионных моделей поверхности волатильности биржевых опционов // Информационные технологии. 2012. № 8. С. 47—52.

УДК 519.854.2:004.023

Е. М. Бронштейн, д-р физ.-мат. наук, проф.,
e-mail: bro-efim@yandex.ru,
Г. А. Карипова, студентка,
Уфимский государственный
авиационный технический университет

Мультиноменклатурная задача транспортной логистики

Рассматривается мультиноменклатурная задача транспортной логистики, в которой учтены возможные запреты на совместную транспортировку некоторых видов грузов. Построена соответствующая линейная частично целочисленная модель. Для решения наряду с точным методом применяется эвристика случайного поиска. Рассмотрены модельные примеры.

Ключевые слова: *маршрутизация, мультиноменклатурность, целочисленное линейное программирование, эвристика*

Введение

Постановка оптимизационных задач транспортной логистики (*Vehicle Routing Problem*, VRP) восходит к работе [1]. Более чем за полвека рассмотрено множество различных задач такого типа, для решения которых применялись как точные, так и эвристические методы (некоторые из многочисленных обзоров [2—7]). В последние годы рассмотрено несколько мультиноменклатурных задач. В частности, в работе [8] предполагалось, что в каждом транспортном средстве (ТС) имеются емкости фиксированной вместимости для перевозки грузов различных видов. В задаче, рассмотренной в статьях [9, 10], учтена невозможность перевозки отдельных видов грузов теми или иными ТС. Решение каждой из таких задач требует модификации методов, разработанных ранее.

В настоящей работе предполагается, что некоторые пары грузов нельзя перевозить совместно. Приведены различные формализации задачи, для решения применяются как точный, так и эвристический метод, основанный на случайном поиске. Приводятся модельные примеры.

Постановка задачи

Пусть имеется один пункт производства (база), несколько пунктов потребления (ПП), грузы различных видов, каждый из которых характеризуется вещественным числом — весом, и парк ТС, каждое из которых может перевозить грузы любых видов, имеет ограничение по весу и характеризуется издержками на единицу пути. Известно, сколько груза каждого вида необходимо доставить в каждый пункт. При этом грузы некоторых видов вместе перевозить нельзя. Ограничения на виды перевозимых грузов могут возникать по разным причинам. Так, например, продукты питания опасно перевозить совместно с химическими или радиоактивными веществами. Задана также матрица (возможно, не симметричная) расстояний между пунктами. После доставки грузов ТС должны возвращаться на базу. Предполагается, что каждое ТС совершает один

цикл. Требуется составить такой план перевозок, при котором суммарные транспортные расходы являются минимальными.

Отметим, что задача не всегда имеет решение: простой случай — сумма грузоподъемностей ТС меньше, чем сумма потребностей во всех пунктах по всем грузам. Иной случай: нельзя перевозить вместе любые виды грузов, а число ТС меньше, чем число видов грузов.

Приведем нелинейную формализацию задачи. Исходные данные:

J — число ПП (пункту производства присваиваем номер 0);

I — число видов перевозимых грузов;

K — число транспортных средств;

Ψ — семейство номеров пар грузов, недопустимых к совместной перевозке;

D^k — грузоподъемность k-го ТС, $k = 1, ..., K$;

B_{ij} — вес груза i-го вида, который необходимо доставить в j-й пункт потребления, $i = 1, ..., I$, $j = 1, ..., J$;

L^k — затраты на проезд k-го ТС на единицу расстояния, $k = 1, ..., K$;

C_{uv} — длина кратчайшего пути от пункта u до пункта v, $u = 0, ..., J$, $v = 0, ..., J$.

Введем обозначения:

x_{ij}^k — вес груза i-го вида, доставляемого k-м ТС в j-й ПП, $i = 1, ..., I$, $j = 1, ..., J$, $k = 1, ..., K$;

$y_i^k = \sum_{j=1}^{J} x_{ij}^k$ — общий вес груза i-го вида, перевозимого k-м ТС, $i = 1, ..., I$, $k = 1, ..., K$;

$c(U)$, где $U \subset \{1, ..., J\}$ — минимальная из длин циклов, содержащих пункты с номерами из U и нулевой пункт;

$F^k = \{j \in \{1, ..., J\} : \sum_{i=1}^{I} x_{ij}^k > 0\}$ — множество пунктов, в которые груз доставляется k-м ТС, $k = 1, ..., K$.

Задачу можно представить следующим образом: найти числа $x_{ij}^k \geqslant 0$ такие, что:

$$\text{Если } (p, q) \in \Psi, \text{ то } y_p^k y_q^k = 0, \ k = 1, ..., K; \quad (1)$$

$$\sum_{k=1}^{K} x_{ij}^k = B_{ij}, \ i = 1, ..., I; \ j = 1, ..., J; \quad (2)$$

$$\sum_{i=1}^{I} \sum_{j=1}^{J} x_{ij}^k \leqslant D^k, \ k = 1, ..., K; \quad (3)$$

$$\sum_{k=1}^{K} c(F^k) L^k \to \min. \quad (4)$$

Условие (1) выражает запрет на совместную перевозку пар грузов из множества Ψ, (2) — это условие обеспечения всех потребителей, (3) — ограничения на вместимость ТС, целевая функция (4) — стоимость перевозок.

Линейная частично целочисленная модель

Ценой существенного повышения размерности задачу удается привести к частично целочисленной линейной форме.

Пусть Z_{ij}^k, $i = 1, ..., I$, $j = 1, ..., J$, $k = 1, ..., K$, — булева переменная, равная 1 тогда и только тогда, когда $x_{ij}^k > 0$. Условие

$$Z_{ij}^k \sum_{k=1}^{K} D^k \geqslant x_{ij}^k \quad (5)$$

обеспечивает выполнение равенства $Z_{ij}^k = 1$ при $x_{ij}^k > 0$, выполнение равенства $Z_{ij}^k = 0$ при $x_{ij}^k = 0$ обеспечивается специальной структурой целевой функции.

Введем булевы переменные Y_j^k ($j = 1, ..., J$; $k = 1, ..., K$), равные 1 тогда и только тогда, когда $\sum_{i=1}^{I} Z_{ij}^k > 0$, т. е. когда k-е ТС доставляет груз в j-й ПП. Это обеспечивается выполнением условий

$$Y_j^k \leqslant \sum_{i=1}^{I} Z_{ij}^k; \quad (6)$$

$$Y_j^k \geqslant Z_{ij}^k \ (i = 1, ..., I). \quad (7)$$

Аналогично введем булевы переменные Q_i^k ($i = 1, ..., I$; $k = 1, ..., K$), равные 1 тогда и только тогда, когда $\sum_{j=1}^{J} Z_{ij}^k > 0$, т. е. когда k-е ТС перевозит груз i-го вида. Это обеспечивается выполнением условий

$$Q_i^k \leqslant \sum_{j=1}^{J} Z_{ij}^k; \quad (8)$$

$$Q_i^k \geqslant Z_{ij}^k \ (j = 1, ..., J). \quad (9)$$

Условие несовместимости перевозимых грузов приобретает следующую форму:

$$Q_p^k + Q_q^k \leqslant 1 \text{ при } (p, q) \in \Psi, \ k = 1, ..., K. \quad (10)$$

Далее введем булевы переменные T_{uv}^k, $u, v = 0, ..., J$; $k = 0, ..., K$, равные 1, если k-е ТС сразу после доставки груза в u-й ПП доставляет груз в v-й. Выполняются следующие условия:

$$\sum_{u=0}^{J} T_{uv}^k = Y_v^k, \ v = 1, ..., J; \ k = 1, ..., K; \quad (11)$$

$$\sum_{v=0}^{J} T_{uv}^k = Y_v^k, \ u = 1, ..., J; \ k = 1, ..., K. \quad (12)$$

Условие (11) означает, что каждое ТС въезжает в каждый ПП, в который оно доставляет груз, один раз. Аналогичное условие (12) относится к выезду из пункта разгрузки.

Отдельно необходимо сформулировать условия на 0-й пункт (базу):

$$T_{00}^k = 0; \quad (13)$$

$$\sum_{v=1}^{J} T_{0v}^k = 1,\ k = 1, \ldots, K; \quad (14)$$

$$\sum_{u=1}^{J} T_{u0}^k = 1,\ k = 1, \ldots, K. \quad (15)$$

Для исключения в маршруте любого ТС циклов, не содержащих базы, воспользуемся модификацией приема, изложенного в ряде источников (например, [11]) для задачи коммивояжера.

Введем целочисленные переменные S_u^k ($k = 1, \ldots, K$; $u = 1, \ldots, J$) такие, что

$$Y_u^k \leqslant S_u^k \leqslant \sum_{j=1}^{J} Y_j^k,\ k = 1, \ldots, K;\ u = 0, \ldots, J; \quad (16)$$

$$0 \leqslant S_u^k \leqslant J Y_u^k,\ k = 1, \ldots, K;\ u = 0, \ldots, J; \quad (17)$$

$$S_u^k - S_v^k + J T_{uv}^k \leqslant J - 1,\ k = 1, \ldots, K;\ u, v = 0, \ldots, J. \quad (18)$$

При выполнении этих условий величины S_u^k равны 0, если k-е ТС не доставляет груз в u-й ПП (17) и, как можно проверить, равно порядковому номеру ПП при движении k-го ТС (16), (18).

Целевую функцию сформируем следующим образом:

$$\sum_{k=1}^{K} \sum_{v=0}^{J} \sum_{u=0}^{J} T_{uv}^k c_{uv} L^k + \max\left\{1, \sum_{k=1}^{K} L^k \sum_{v=0}^{J} \sum_{u=0}^{J} c_{uv}\right\} \sum_{k=1}^{K} \sum_{i=1}^{I} \sum_{j=1}^{J} Z_{ij}^k. \quad (19)$$

Поясним смысл целевой функции. Первое слагаемое равно транспортным издержкам. Второе слагаемое обеспечит равенство 0 величин Z_{uv}^k при $x_{uv}^k = 0$.

Разумеется, линейные условия (2), (3) сохраняются.

Таким образом, линейная модель описывается следующим образом. Найти вещественные числа $x_{ij}^k \geqslant 0$, булевы переменные Z_{ij}^k, Y_j^k, Q_i^k, T_{uv}^k, целочисленные переменные S_u^k, для которых выполняются условия (2), (3), (5)—(19).

В задаче (1)—(4) число неизвестных равно IJK, число ограничений — $IJK + IJ + K + KR$, где R — число пар грузов, запрещенных к совместной перевозке.

В линейной частично целочисленной задаче (2), (3), (5)—(19) число переменных равно $K(2IJ + I + 2J + (J + 1)^2)$, число ограничений $K(4IJ + I + 7J + (J + 1)^2 + R + 3)$. Тем самым, наибольшее влияние на размерность задачи (2), (3), (5)—(19) оказывает число ПП.

Численные методы решения задачи

Для решения задачи (2), (3), (5)—(19) применялся стандартный пакет линейной оптимизации LPSolveIDE 5.5.2.0.

К нелинейной задаче (1)—(4) применялся эвристический метод, основанный на случайном поиске. Приведем его описание.

1. Принятие решения о загрузке всех ТС, удовлетворяющих ограничениям на грузоподъемность ТС, на совместную перевозку грузов и суммарному условию удовлетворения потребностей во всех пунктах потребления (редуцированная задача). Для этого решается вспомогательная задача линейного частично целочисленного программирования (меньшей размерности по сравнению с (2), (3), (5)—(19)), в которой искомыми являются загрузки ТС по видам грузов (но не по пунктам потребления!).

Приведем формулировку вспомогательной задачи.

Введем следующие обозначения.

$P_i = \sum_{j=1}^{J} B_{ij}$ ($i = 1, \ldots, I$) — суммарная потребность в грузе i-го вида во всех ПП;

r_i^k ($i = 1, \ldots, I$; $k = 1, \ldots, K$) — вес груза i-го вида, перевозимый k-м ТС.

Ограничения редуцированной задачи аналогичны рассмотренным выше и имеют вид:

$$r_i^k \geqslant 0; \quad (20)$$

$$\sum_{k=1}^{K} r_i^k = P_i; \quad (21)$$

$$r_p^k r_q^k = 0 \text{ при } (p, q) \in \Psi; \quad (22)$$

$$\sum_{i=1}^{I} r_i^k \leqslant D^k. \quad (23)$$

Здесь обозначения Ψ, D^k имеют тот же смысл, что и в задаче (1)—(4).

Система ограничений (20)—(23) в общем случае может иметь множество решений. Для выделения какого-либо решения для последующих этапов работы алгоритма введем дополнительно линейную целевую функцию

$$\sum_{k=1}^{K} \sum_{i=1}^{i} \mu_i^k r_i^k \to \min. \quad (24)$$

Коэффициенты $\mu_i^k \in [-1, 1]$ линейной формы генерируются датчиком случайных чисел.

С помощью приемов, использованных при построении задачи (2), (3), (5)—(19), удается перейти от нелинейной задачи (20)—(24) к задаче частично целочисленного линейного программирования. Полученная линейная задача также решалась с помощью пакета LPSolvelDE 5.5.2.0.

2. Построение маршрутов ТС с помощью жадного алгоритма. Для этого ТС упорядочивались по убыванию загруженности и каждое ТС последовательно направлялось в ближайший пункт, в котором есть потребность в грузе, который в этот момент имеется в ТС.

3. Шаги 1, 2 повторялись 100 раз. Если минимальное значение суммарной протяженности маршрутов достигалось при первых 90 итерациях, то работа алгоритма завершалась. В противном случае алгоритм запускался заново (разумеется, с запоминанием достигнутого рекорда). Общее число повторений шагов 1, 2 ограничивалось 1000.

Вычислительный эксперимент

Эксперимент проводился для задач небольшой размерности.

Исходные данные:

- пункт производства — г. Уфа;
- пункты потребления — гг. Казань, Челябинск, Пермь, Екатеринбург, Самара (им присвоены номера 1—5 в указанном порядке).

Грузы необходимо доставить с помощью трех ТС марки MAN с характеристиками: грузоподъемность первого и второго 20 т, удельные транспортные издержки 7,5 руб./км, для третьего эти характеристики соответственно 24 т и 8,5 руб./км.

Имеются грузы четырех видов. Запрещены к совместной перевозке грузы первого и второго; третьего и четвертого видов. Потребности в грузах (в тоннах) заданы табл. 1.

Матрица кратчайших расстояний (в км) между пунктами симметрическая, ее верхняя треугольная часть имеет вид табл. 2.

Решение задачи (2), (3), (5)—(19) представлено в виде табл. 3—5.

Маршрут первого ТС имеет вид 0-2-4-3-1-5-0, второго — 0-4-3-0, третьего совпадает с первым. Суммарные затраты на перевозку составляют 48 823 руб.

При применении эвристического метода рекорд был достигнут на 61 итерации. Планы перевозок приведены в табл. 6—8.

Маршруты первого и второго ТС совпадают с маршрутами первого и третьего ТС, полученными точным методом (ТС проезжают все города), маршрут третьего ТС имеет вид: 0-2-4-3-0. Суммарные затраты составляют 48 944,5 руб. (превосходит точный результат на 121,5 руб. или 0,25 %). При этом решение эвристическим методом на ПК с 4-ядерным процессором Intel i5, тактовая частота 2,90 ГГц было получено за 2 ч, а точным — за 31 ч.

Таблица 1

Потребности в грузах

i \ j	1	2	3	4	5
1	2	2	5	5	0
2	1	5	4	3	1
3	3	3	5	4	3
4	2	4	3	1	2

Таблица 2

Расстояния между пунктами

	0	1	2	3	4	5
0	0	534	412	465	476	458
1	—	0	946	660	1017	343
2	—	—	0	570	213	870
3	—	—	—	0	357	833
4	—	—	—	—	0	934
5	—	—	—	—	—	0

Следует отметить, что хотя пакет LPSolve формально не имеет ограничений на размерность, поиск решения может занять длительное время, как в приведенном примере, или даже не привести к результату (см. далее).

Полученные результаты позволяют сделать вывод о состоятельности разработанного эвристического метода. Естественно предположить, что данный метод окажется пригодным и для задач большей размерности.

Таблица 3

План перевозок для первого ТС (точный метод)

	1	2	3	4	5
1	2	2	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	2	4	3	1	2

Таблица 4

План перевозок для второго ТС (точный метод)

	1	2	3	4	5
1	0	0	5	5	0
2	0	0	0	0	0
3	0	0	5	4	0
4	0	0	0	0	0

Таблица 5

План перевозок для третьего ТС (точный метод)

	1	2	3	4	5
1	0	0	0	0	0
2	1	5	4	3	1
3	3	3	0	0	3
4	0	0	0	0	0

Таблица 6

План перевозок для первого ТС (эвристический метод)

	1	2	3	4	5
1	0	0	0	0	0
2	1	5	4	3	1
3	3	0	0	0	3
4	0	0	0	0	0

Таблица 7

План перевозок для второго ТС (эвристический метод)

	1	2	3	4	5
1	2	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	2	4	3	1	2

Таблица 8

План перевозок для третьего ТС (эвристический метод)

	1	2	3	4	5
1	0	2	5	5	0
2	0	0	0	0	0
3	0	3	5	4	0
4	0	0	0	0	0

Таблица 9

Потребности в грузах

	1	2	3	4	5	6
1	2	3	5	5	0	2
2	1	5	6	4	3	3
3	3	4	5	6	4	4
4	4	6	3	2	2	2

Таблица 10

Расстояния между пунктами

	0	1	2	3	4	5	6
0	0	534	412	465	476	458	372
1	—	0	946	660	1017	343	728
2	—	—	0	570	213	870	738
3	—	—	—	0	357	833	837
4	—	—	—	—	0	934	848
5	—	—	—	—	—	0	440
6	—	—	—	—	—	—	0

Приведем пример решения задачи большей размерности с помощью эвристического метода.

Исходные данные:

- пункт производства — г. Уфа;
- пункты потребления — гг. Казань, Челябинск, Пермь, Екатеринбург, Самара, Оренбург (им присвоены номера 1—6 в указанном порядке).

Грузы необходимо доставить с помощью четырех ТС марки MAN со следующими характеристиками: грузоподъемность первого и второго 20 т, удельные транспортные издержки 7,5 руб./км, для третьего и четвертого эти характеристики соответственно 24 т и 8,5 руб./км.

Необходимо доставить груз четырех видов. Запрещены к совместной перевозке грузы первого и второго; третьего и четвертого видов. Потребности в грузах (в тоннах) заданы табл. 9.

Матрица кратчайших расстояний (в км) между пунктами симметрическая, ее верхняя треугольная часть приведена в табл. 10.

При применении эвристического метода рекорд был достигнут на 38 итерации. Планы перевозок приведены в табл. 11—14.

Маршрут первого ТС имеет вид 0-2-4-3-0, второго 0-6-5-1-3-4-0, третьего 0-6-1-3-4-2-0, четвертого 0-6-5-1-3-4-2-0. Суммарные затраты составляют 77 794 руб. Примерное время, затраченное на обработку входных данных и получение решения, составляет 2,5 ч. Точным методом за приемлемое время решение получить не удалось.

Заключение

Рассмотрена задача маршрутизации с новыми естественными ограничениями на совместную перевозку грузов разных видов. Приведены нелинейная и линейная частично целочисленная формализации. Для нелинейной задачи разработан эвристический алгоритм. Проведенный численный эксперимент показал, что эвристический метод дает решение за приемлемое время, точный алгоритм неприменим даже для небольших размерностей.

Работа выполнена при поддержке РФФИ (проект 10-06-00001).

Таблица 11

План перевозок для первого ТС

	1	2	3	4	5	6
1	0	0	0	0	0	0
2	0	5	5	4	0	0
3	0	4	0	2	0	0
4	0	0	0	0	0	0

Таблица 12

План перевозок для второго ТС

	1	2	3	4	5	6
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	3	0	5	4	4	4
4	0	0	0	0	0	0

Таблица 13

План перевозок для третьего ТС

	1	2	3	4	5	6
1	2	3	5	5	0	2
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	3	0	0	0	0

Таблица 14

План перевозок для четвертого ТС

	1	2	3	4	5	6
1	0	0	0	0	0	0
2	1	0	1	0	3	3
3	0	0	0	0	0	0
4	4	3	3	2	2	2

Список литературы

1. **Dantzig G. B., Ramser J. H.** The Truck Dispatching Problem // Management Science. 1959. V. 6 (1). P. 80—91.

2. **Parragh S., Doerner K., Hartl R.** A survey on pickup and delivery problems. Part I: Transportations between customers and depot // Journal of Betriebswirtschaft, 2008. V. 58. P. 21—51.

3. **Parragh S., Doerner K., Hartl R.** A survey on pickup and delivery problems. Part II: Transportations between customers and depot // J. Betriebswirtschaft. 2008. V. 58, N 2. P. 81—117.

4. **Berbeglia G., Cordeau J. F., Gribkovskaia I., Laporte G.** Static pickup and delivery problems: A classification scheme and survey // TOP. 2007. V. 15, N 1. P. 1—31.

5. **Laporte G.** Metaheuristics for the Vehicle Routing Problem: Fifteen Years of Research Canada Research Chair in Distribution Management HEC. Montreal. 2007.

6. **Van Hoeve W.-J.** Constraint-based solution methods for vehicle routing problems. Tepper School of Business, Carnegie Mellon University, EWO Seminar. 2009. November 17.

7. **Toth P. and Vigo D.** The Vehicle Routing Problem. Philadelphia. 2002. SIAM.

8. **Mayldermans L., Pang G.** On the be benefits of co-collection: Experiments with a multi-compartment vehicle routing algorithm // European Journal of Operational Research. 2010. N 206. P. 93—103.

9. **Бронштейн Е. М., Заико Т. А.** Оптимизационные задачи транспортной логистики // Автоматика и телемеханика. 2010. № 10. С. 133—147.

10. **Бронштейн Е. М., Заико Т. А.** Задача маршрутизации с запретами // Информационные технологии. 2011. № 6. С. 42—44.

11. **Gavish B.** A note on the formulation of the *m*-salesman traveling salesman problem // Management Science. 1976. T. 22 (6). P. 704—705.

УДК 519.2:622

В. А. Антонов, д-р техн. наук, глав. науч. сотр.,
Институт горного дела УрО РАН,
e-mail: antonov@igduran.ru

Построение и оптимизация моделей нелинейной функционально-факторной регрессии

Изложена методология формирования, оптимизации и распространения моделей нелинейной функционально-факторной регрессии. Функции, входящие в уравнения регрессионных моделей, задаются как математические выражения факторов влияния исследуемых объектов или процессов на искомую регрессию и изначально представляются в общем виде. Конкретные значения внутренних параметров функций рассчитывают в области дробных рациональных чисел в ходе оптимизации уравнений по критерию максимума коэффициента их детерминации. Показаны методические приемы оптимизации специально разработанным методом приближений параболической вершины. Приведены примеры применения методологии в экспериментальных исследованиях.

Ключевые слова: *модель регрессии, факторные функции, функциональные параметры, оптимизация, программа для ЭВМ*

Введение

Регрессионные модели создают для интерпретации результатов измерений физических и других величин, полученных в ходе экспериментальных исследований свойств естественных и техногенных объектов, технологических, биологических и социально-экономических процессов. Искомые закономерности, по которым они образуются, развиваются и изменяются, часто бывают настолько сложными, что достоверно их описать можно только уравнениями нелинейной регрессии, требующими оценивания внутренних параметров функций, входящих в уравнения. Однако расчет таких параметров в настоящее время весьма затруднен или вовсе невозможен ввиду ограничений применения и неустойчивой сходимости известных алгоритмов их оптимизации.

Для решения обозначенной проблемы в Институте горного дела УрО РАН автором разработан новый и относительно простой численный оптимизационный метод приближений параболической вершины (МППВ) [1, 2]. Совместное применение МППВ и метода наименьших квадратов (МНК) обеспечивает устойчивый расчет параметров функций, составляющих уравнение. Такими параметрами могут быть, например, показатели степени, коэффициенты в показателях степени, основания и т. д. Теперь стало возможным функции уравнений задавать в общем виде и рассматривать их как математическое выражение факторов влияния на регрессию природных или каких-либо иных процессов. В связи с численными расчетами оптимальных значений параметров функций, составляющих уравнения, они названы функционально-факторными с самоопределяющимися параметрами (ФСП). В плане развития методологии создания таких регрессионных моделей и их практического применения представляются актуальными изложенные ниже аспекты формирования и распространения уравнений ФСП, включая постановку цели регрессии, выбор ее интервалов с учетом действующих факторных функций, оптимизацию полученных уравнений на основе МППВ и МНК.

Цели, факторы и интервалы регрессии

Под фактором понимается причина, исходящая от какого-либо объекта, процесса или явления, влияющая на регрессию и определяющая ее характер или отдельные черты. Действие фактора зависит от переменных величин, т. е. аргументов, и выражается математической функцией, общий вид которой формируется по теоретическим представлениям факторного влияния, объясняющим регрессию. Число факторов так же, как число аргументов, может быть разным.

Итоговые значения параметров, устанавливающие окончательный и конкретный вид факторных функций по степени возрастания или убывания, вогнутости или выпуклости, объективно рассчитываются МППВ как оптимальные в области рациональных дробных, положительных и отрицательных чисел. Поэтому уравнения ФСП распространяются в широких интервалах интерполяции и экстраполяции со значительно большей достоверностью по сравнению с известными уравнениями, в которых функции и функциональные параметры, например показатели степени, назначают искусственно. Это обстоятельство значительно расширяет возможности регрессии и позволяет с помощью данных уравнений достичь следующих целей.

- Установить в конкретном виде и объяснить математически закономерность, по которой с некоторым случайным отклонением распределены значения зависимой величины, полученные в экспериментах.
- Создать математическую регрессионную модель исследуемых объектов или процессов.
- Определить характерные параметры, например, показатели степени, координаты экстремумов, особых точек, положение асимптот, пересечения с осями координат в математических моделях отображаемых регрессией объектов или процессов.

- Предсказать значения зависимой величины (дать прогноз) в областях интерполяции и экстраполяции.
- Управлять объектом или технологическим процессом по уравнению регрессии.

Для получения результата, заданного целью, интервалы регрессии распространяются по аргументам на соответствующую область. Распространение возможно лишь при условии, что данная область и окрестности узловых точек, по которым строят регрессию, однородны по факторным и другим влияющим на регрессию свойствам. Таким образом, цели, факторы и интервалы регрессии взаимосвязаны. Поэтому принимаем, что в областях интерполяции и экстраполяции так же, как в экспериментально полученных узловых точках, действуют факторы, функционально учтенные в уравнениях, и не предполагается появление других, вновь возникших, факторов. Кроме того, при удалении от узловых точек в этих областях сохраняются другие свойства пространства, косвенно влияющие на регрессию. Новые свойства пространства в них возникать не должны.

Внешние границы распространения регрессионной модели устанавливаются там, где по теоретическим соображениям прекращается действие хотя бы одного из учтенных факторов. Это может произойти на краю зоны локального распространения факторов, при появлении обусловленного чем-либо граничного значения зависимой величины или факторной функции, например, смене их алгебраического знака, превышении заданного уровня.

Начальное формирование модельных уравнений

Уравнения регрессии представляются полиномами, в которых содержатся постоянная составляющая (свободный член) и одна или несколько факторных функций. Их общий вид, исходя из полноты теоретического объяснения регрессии, может быть простым и сложным. Наиболее простые функции с одним аргументом, например степенные или показательные, выражают факторное влияние, проявляющееся положением заданных узловых точек на участках регрессии с разной монотонностью. Если факторы действуют аддитивно (независимо), то функции суммируются. При мультипликативном (зависимом) действии функции умножаются. Часто факторы влияют комбинированно, т. е. аддитивно и мультипликативно. При наличии двух и большего числа аргументов монотонности убывания и возрастания регрессии могут быть направлены вдоль их координатных осей и под углом к ним. Соответственно, различают факторные функции осевые и диагональные. Функции осевые зависят от одного, а диагональные — от нескольких аргументов. В качестве примера факторного формирования модели регрессии представим трехмерное уравнение

$$y = A_1 x_1^{\mu_1} + A_2 \mathbf{e}^{\beta x_2} + A_3 x_1^{\mu_{1m}} x_2^{\mu_{2m}} + B,$$

в котором монотонности изменения y, направленные вдоль координатных осей аргументов x_1 и x_2, объясняются аддитивным влиянием осевых факторов, выраженных, соответственно, степенной и показательной функциями. Дополнительная монотонность y, направленная под углом к координатным осям x_1 и x_2, обусловлена влиянием диагонального фактора, выраженного произведением степенных функций соответствующих координат. В приведенном уравнении коэффициенты A и B оптимизируются МНК, а параметры μ_1, β, μ_{1m}, μ_{2m} — МППВ.

Оптимизация уравнений

Критерием оптимизации уравнений регрессии является достижение максимума целевой функции — коэффициента их детерминации $R^2(\xi_j)$, где ξ_j — параметры факторных функций. Если распределение $R^2(\xi_j)$ имеет два или несколько глобальных максимума, то оптимизационные расчеты МППВ проводятся в последовательно чередующихся его ортогональных сечениях, содержащих целевые функции $R^2_{\text{о}}$. Каждое сечение $R^2_{\text{о}}$ образуется путем задания в функции R^2 фиксированных значений некоторым параметрам ξ так, что в оставшихся координатах $\xi_1, \xi_2, ..., \xi_j, ..., \xi_m$ оно содержит единственный глобальный максимум $R^2_{\text{ом}}$. Достаточным условием выбора сечения является единственный минимум суммы квадратов остатков, образующихся в системе его аргументов ξ_j, при сравнении регрессии с заданными в узловых точках значениями зависимой величины. Приближенно единственность глобального максимума оценивается также существованием лишь одной монотонности и однозначности факторной функции при изменении значений переменных ξ_j выбранного сечения.

Суть оптимизации МППВ состоит в итерационном последовательном приближении максимума вогнутой параболической функции, построенной по опорным точкам, принадлежащим множеству $R^2_{\text{о}}$, к максимуму $R^2_{\text{м}}$. В каждом приближении проводится замена опорной точки с наименьшим значением $R^2_{\text{о}\nu}$ на точку, рассчитанную по вершине параболической функции. В данном разделе более детально изложены некоторые важные особенности реализации метода, выполнение которых обеспечивает его устойчивую сходимость.

Во всех сечениях проводится выделение, т. е. локализация обозначенного выше глобального максимума. По результатам локализации устанавливаются стартовые координаты $\xi_{j\nu}$ и $R^2_{\text{о}\nu}$ опорных точек, которые используются в МППВ. Здесь индекс j

пробегает значения от 1 до m — числа оптимизируемых в сечении параметров, а индекс ν обозначает номер опорной точки и изменяется от 1 до $2m + 1$. В сечениях, содержащих аддитивно или мультипликативно взаимодействующие функции, применяется функциональная и, соответственно, осевая или центральная локализация. Начальное положение опорных точек в функциональной локализации определяется аналитически по формулам с учетом характерных особенностей распределения значений зависимой величины, по которым строится регрессия. В осевой локализации проводится сканирование R_o^2 вдоль профилей, расположенных параллельно осей ξ_j. По обнаруженным осевым максимумам R_{oj}^2 определяется координата первой опорной точки с номером $\nu = 1$. В центральной локализации сканирование R_o^2 проводится сначала в полярной системе по углу вокруг начала координат аргументов ξ_j. Затем выделяется луч с наиболее вероятным направлением на область, содержащую максимум $R_{ом}^2$, и вдоль его проводится дополнительное сканирование R_o^2. По обнаруженному на луче максимуму R_o^2 устанавливается координата опорной точки также с единичным номером.

В качестве примера на рис. 1 (см. третью сторону обложки) показаны схемы осевой и центральной локализации, проводимой в сечениях, содержащих по два аргумента ξ_1 и ξ_2. В осевой локализации (рис. 1, *а*) опорная точка с номером 1 установлена на пересечении отсчетов профильных максимумов коэффициента детерминации R_o^2. В центральной локализации (рис. 1, *б*) по результатам сканирования коэффициента R_o^2 под разным полярным углом φ определено направление на область локализации, обозначенное штриховым лучом. Первая опорная точка установлена в месте лучевого максимума коэффициента детерминации.

Стартовое положение остальных опорных точек с номерами $\nu > 1$ задается по следующему правилу ортогонального вогнутого перекрестка, которое состоит из двух частей. Центром перекрестка считаем положение первой опорной точки с номером $\nu = 1$. Аргументы $\xi_{j\nu}$ опорных точек с номерами $1 < \nu < 2j$ и $2j + 1 < \nu \leqslant 2m + 1$ тоже принимают значения центра. Опорные точки с номерами $2j$ и $2j + 1$ устанавливаются по оси аргумента j смещенными относительно центра в двух противоположных направлениях на интервал $|\Delta\xi_j|$. Координаты точек с номерами $\nu = 2j$ выражаются разностью

$$\xi_{j\nu} = \xi_{j1} - |\Delta\xi_j|,$$

а с номерами $\nu = 2j + 1$, соответственно, суммой

$$\xi_{j\nu} = \xi_{j1} + |\Delta\xi_j|.$$

Координаты опорных точек, установленные по первой части изложенного правила, приведены в табл. 1. Интервал $|\Delta\xi_j|$ в стартовом состоянии опорных точек назначается намного меньшим, чем размер области локализации.

Таблица 1

Координаты опорных точек МППВ, установленные по первой части правила ортогонального вогнутого перекрестка

ν	$\xi_{1\nu}$	$\xi_{2\nu}$	$\xi_{3\nu}$	...	$\xi_{m\nu}$
1	ξ_{11}	ξ_{21}	ξ_{31}	...	ξ_{m1}
2	$\xi_{11} - \|\Delta\xi_1\|$	ξ_{21}	ξ_{31}	...	ξ_{m1}
3	$\xi_{11} + \|\Delta\xi_1\|$	ξ_{21}	ξ_{31}	...	ξ_{m1}
4	ξ_{11}	$\xi_{21} - \|\Delta\xi_2\|$	ξ_{31}	...	ξ_{m1}
5	ξ_{11}	$\xi_{21} + \|\Delta\xi_2\|$	ξ_{31}	...	ξ_{m1}
6	ξ_{11}	ξ_{21}	$\xi_{31} - \|\Delta\xi_3\|$	...	ξ_{m1}
7	ξ_{11}	ξ_{21}	$\xi_{31} + \|\Delta\xi_3\|$	...	ξ_{m1}
...	...	...	...	...	...
$2m$	ξ_{11}	ξ_{21}	ξ_{31}	...	$\xi_{m1} - \|\Delta\xi_m\|$
$2m + 1$	ξ_{11}	ξ_{21}	ξ_{31}	...	$\xi_{m1} + \|\Delta\xi_m\|$

Распределение R_o^2 в области локализации максимума аппроксимируется в МППВ следующей параболической функцией $P(\xi)$:

$$P = C + \sum_{j=1}^{m} (C_{j1}\xi_j + C_{j2}\xi_j^2).$$

Название вогнутого ортогонального перекрестка опорных точек отражает то обстоятельство, что сечения параболической функции, проведенные вдоль каждого j-аргумента через центр перекрестка по трем опорным точкам с координатами

$$\xi_{j1} - |\Delta\xi_j|,\ \xi_{j1},\ \xi_{j1} + |\Delta\xi_j|,$$

представляют собой вогнутые параболы. Они расположены во взаимно перпендикулярных плоскостях и имеют общий максимум $P_в$. При этом должны выполняться неравенства $C_{j2} \leqslant 0$. При соответствующей вогнутости функции R_o^2 в области локализации и гладкой ее топологии установка опорных точек по данному правилу обеспечивает продвижение итераций в МППВ.

Возможно, что топология функции R_o^2 в области перекрестка имеет осложнения, например, локальный склон, перегиб. Поэтому одно или несколько упомянутых неравенств могут не выполняться. Тогда по второй части правила перекрестка проводится его расширение вдоль аргументов таких неравенств. Опорные точки с номерами $2j$ и $2j + 1$ смещаются по j-му аргументу относительно центра перекрестка в направлении, которое определяется по результатам сравнения значений R_o^2 в точках $\xi_{j1} - |\Delta\xi_j|$ и $\xi_{j1} + |\Delta\xi_j|$. Расширение проводится по направлению, проходящему от центра через точку, имеющую наибольшее значение R_o^2. Эти точки перемещаются поочередно с опережением друг друга, т. е. "перепрыгиванием" одной точки через другую с ус-

тановленным интервалом. Процедура расширения прекращается после восстановления соответствующих неравенств $C_{j2} < 0$ или при достижении координат перемещающихся опорных точек предельно допустимых значений.

После установки опорных точек по изложенному правилу проводят итерации всего перекрестка или отдельных его точек в следующем порядке.

Используя координаты опорных точек перекрестка, рассчитывают по формуле

$$\xi_{j\text{в}} = -C_{j1}/2C_{j2}$$

совокупность вершинных аргументов $\xi_{j\text{в}}$ параболической функции. Затем, принимая их значения и рассчитанное с их учетом значение функции $R^2_{\text{ов}}(\xi_{j\text{в}})$, получают вновь образованную опорную точку. После этого возможны следующие три случая с разным действием.

В первом случае получено значение $R^2_{\text{ов}}$ меньше наименьшего R^2_{ov} в опорных точках. Среди их номеров $\nu > 1$ находится точка с наибольшим отсчетом R^2_{ov} и большим, чем отсчет R^2_{o1} в центре. Тогда проводится итерация перекрестка. Этой точке присваивается номер 1 и вокруг нее строится новый перекресток, т. е. происходит перенос перекрестка на некоторый интервал в направлении роста R^2_{ov}. В случае, когда наибольший отсчет R^2_{ov} меньше, чем отсчет R^2_{o1} в центре, интервалы $|\Delta\xi_j|$ перекрестка увеличивают. Если после такого увеличения соотношение отсчетов в опорных точках не изменилось, то итерации в данном сечении пропускаются, поскольку центральная точка является приближенным максимумом R^2_{o}.

Во втором случае получено значение $R^2_{\text{ов}}$ больше наибольшего значения R^2_{ov} в опорных точках. Тогда так же проводят итерации перекрестка. Его центр перестанавливается (переносится) на место новой опорной точки с координатами $\xi_{j\text{в}}$, $R^2_{\text{ов}}$. Теперь этой точке присваивается номер один. Остальные опорные точки устанавливают вокруг нее по уже упомянутому правилу перекрестка. При этом коэффициент детерминации в опорных точках возрастает.

В третьем случае получено значение $R^2_{\text{ов}}$ больше наименьшего и меньше наибольшего значения R^2_{ov} в опорных точках. Тогда проводят итерации опорных точек. Заменяют одну из них с наименьшим значением R^2_{ov} на вновь образованную опорную точку.

В каждом следующем k-м приближении опорных точек, используя их координаты, рассчитывают координаты вновь образованной точки

$$\xi_{j\text{в}k} = -C_{j1k}/2C_{j2k} \text{ и } R^2_{\text{ов}k}(\xi_{j\text{в}k}).$$

В $(k + 1)$-м приближении действуют по следующим условиям.

Если значение $R^2_{\text{ов}k}$ превышает $R^2_{\text{ov}k}$ хотя бы в одной опорной точке и выполнено хотя бы одно соотношение $C_{j2} < 0$, то заменяют одну из опорных точек с наименьшим значением $R^2_{\text{ov}k}$ на вновь образованную опорную точку.

Если $R^2_{\text{ов}}$ по значению равно или меньше наименьшего из R^2_{ov} или нарушены все неравенства $C_{j2}k < 0$, то итерации временно останавливают и проводят коррекцию координат опорных точек. Их устанавливают по тому же правилу ортогонального вогнутого перекрестка (с возможным его расширением), но с учетом достигнутого на момент коррекции итерационного приближения. Теперь центром перекрестка считается единственная сохраняемая опорная точка, имеющая наибольший отсчет R^2_{ov}. Ей присваивается номер ν, равный единице. Остальные опорные точки смещаются относительно центра на интервал $|\Delta\xi_j|$, который в данном случае представляется усредненным текущим отклонением опорных точек k-го приближения от их взвешенного центра:

$$|\Delta\xi_{jk}| = \frac{1}{t}\sum_{\nu=1}^{\nu=t}|\xi_{j\nu k} - \xi_{j\text{ц}k}|,$$

где $\xi_{j\text{ц}k}$ — j-я координата взвешенного центра опорных точек, определяемая выражением

$$\xi_{j\text{ц}k} = \frac{\xi_{j1k}R^2_{1k} + \xi_{j2k}R^2_{2k} + \ldots + \xi_{jtk}R^2_{tk}}{R^2_{1k} + R^2_{2k} + \ldots + R^2_{tk}}.$$

Коррекция опорных точек считается завершенной, когда восстанавливаются все неравенства $C_{j2k} < 0$. Далее итерации возобновляют по указанному выше порядку перемещения ортогонального перекрестка или отдельных опорных точек в направлении роста коэффициента детерминации.

Выполнение указанного порядка итераций и правил приводит к сходимости МППВ. Максимум $P_{\text{в}}$ параболической функции, рассчитанный в условиях $C_{j2k} < 0$, будет всегда не меньше наибольшего значения R^2_{ov} в опорных точках. Поскольку целевая функция R^2_{o} в области локализации ее максимума выпукла вверх, и в каждой итерации "крайняя" опорная точка с наименьшим значением R^2_{ov} заменяется расчетной вершинной точкой параболической функции, то по мере увеличения числа итераций значения R^2_{ov} во всех опорных точках дискретно возрастают. Влияние местных нарушений в выпуклости целевой функции устраняется управляемым переносом и коррекцией положения опорных точек. При этом центр перекрестка, где они устанавливаются, располагается все ближе к точке, соответствующей максимуму $R^2_{\text{ом}}$. В ходе итераций область опорных точек и размер $|\Delta\xi_j|$ корректирующего перекрестка уменьшаются. Опорные точки концентрируются, и качество аппроксимации области максимума функции R^2_{o} вершиной параболической функции улучшается. Совокупность аргументов

$\xi_{1вk}$, $\xi_{2вk}$, ..., $\xi_{mвk}$, рассчитанных по вершине параболической функции, все меньше отличается от совокупности искомых аргументов, соответствующих максимуму функции R^2_{o}, т. е. опорные точки устремляются к предельному значению функции ортогонального сечения — максимуму $R^2_{ом}$. При смене сечений рассчитанные в них значения $R^2_{ом}$ также все меньше отличаются, приближаясь при этом к искомому максимуму $R^2_{м}$.

Достаточность приближений внутри сечения оценивается сравнением сколь угодно малой допустимой, т. е. заданной, простой средней относительной погрешности δ определения вершинных параметров ξ_j с ее частным значением δ_{jk}, рассчитанным в текущем приближении для каждого аргумента по формуле

$$\delta_{jk} = \frac{\sum_{f=1}^{t} |\xi_{j(k+1)} - \xi_{jkf}|}{t|\xi_{j(k+1)}|},$$

где ξ_{jkf} — значение j-го аргумента опорной f-й точки в k-м приближении; $\xi_{j(k+1)}$ — значение j-го аргумента новой опорной точки, рассчитанной для приближения $(k + 1)$. Приближения с номером k считаются достаточными, если для каждого аргумента выполняется неравенство

$$\delta_{jk} \leqslant \delta.$$

Достаточность чередования сечений определяется по сходимости рассчитанных в них параметров ξ_j. Сходимость оценивается по нескольким последним результатам расчетов, повторяющихся в очереди, когда в полученных малых приращениях $\Delta\xi_j$, соответствующих погрешности δ, появляются разные алгебраические знаки.

Методики оптимизации уравнений кратко обозначаются аббревиатурой, включающей сведения об оптимизации в сечениях R^2_{o}, входящих в один цикл их чередования. В методике (М) указывается: способ локализации максимума функции R^2_{o} (ОЛ — осевая, ЦЛ — центральная); число опорных точек, по которым строится параболическая функция (3Т, 5Т и т. д.); число кратности применения сечения (проставляется через тире). Например, аббревиатура МЦЛ5Т-1, МОЛ3Т-2 обозначает методику оптимизации, проводимую по трем сечениям. В сечении с пятью опорными точками для определения их центра на старте итераций применена центральная локализация R^2_{o}. Оптимизируются два параметра уравнения. В двух сечениях с тремя опорными точками области вершины R^2_{o} на старте итераций выделены осевой локализацией. В каждом из них оптимизируется по одному параметру. Всего по результатам оптимизации рассчитываются четыре параметра.

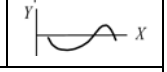
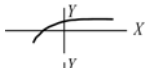
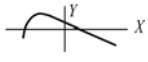
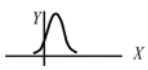
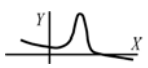
Практическое построение регрессионных моделей

Построение регрессионных моделей разной мерности по приведенной методологии покажем на следующих примерах.

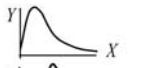
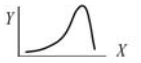
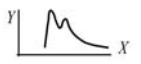
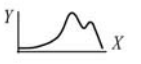
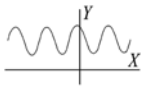
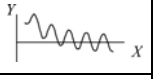
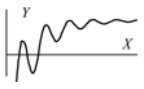
Построение двумерных моделей по серии уравнений ФСП реализовано в компьютерной программе для ЭВМ "Уравнения нелинейной регрессии, тренды двумерные функционально-факторные с самоопределяющимися параметрами и повышенной достоверностью (Тренды ФСП-1)" [3], разработанной в Институте горного дела УрО РАН. Перечень программных уравнений приведен в табл. 2. В уравнениях ПС СПС аддитивное влияние от одного до трех регрессионных факторов выражено функциями степенными, а в уравнениях ПП СК влияние одного или двух факторов — функциями показательными. В уравнении ППЛС СК совместно действуют факторы, выраженные функциями показательной

Таблица 2

Общий вид уравнений регрессии, оптимизируемых программой "Тренды ФСП-1"

Тип уравнения (самоопределяющиеся параметры)	Формула уравнения	Характерный график
Полиномные степенной функции с самоопределяющимися показателями степени μ (ПС СПС), $X > 0$		
A − $X(\mu)$	$Ax^{\mu} + B$	
A − $X(\mu_1, \mu_2)$	$A_1 x^{\mu_1} + A_2 x^{\mu_2} + B$	
A − $X(\mu_1, \mu_2, \mu_3)$	$A_1 x^{\mu_1} + A_2 x^{\mu_2} + A_3 x^{\mu_3} + B$	
Полиномные показательной функции с самоопределяющимися коэффициентами β в показателях степени (ПП СК)		
A − $X(\beta)$	$Aa^{\beta x} + B$	
A − $X(\beta_1, \beta_2)$	$A_1 a^{\beta_1 x} + A_2 a^{\beta_2 x} + B$	
Полиномное показательной и линейной степенной функций с самоопределяющимся коэффициентом β в показателе степени (ППЛС СК)		
A − $X(\beta, \mu = 1)$	$A_1 a^{\beta x} + A_2 x + B$	
Нормального распределения с самоопределяющимися смещением $\bar{x}$ и коэффициентом σ (НР СК), σ > 0		
M − $X(\bar{x}, \sigma)$	$A\frac{1}{\sigma\sqrt{2\pi}}\mathbf{e}^{-\frac{(x-\bar{x})^2}{2\sigma^2}}$	
Полиномное показательной и нормально распределенной функций с самоопределяющимися смещением $\bar{x}$ и коэффициентами β, σ (ППНР СК), σ > 0		
AM − $X(\beta, \bar{x}, \sigma)$	$A_1\mathbf{e}^{\beta x} + A_2\frac{1}{\sigma\sqrt{2\pi}}\mathbf{e}^{-\frac{(x-\bar{x})^2}{2\sigma^2}} + B$	

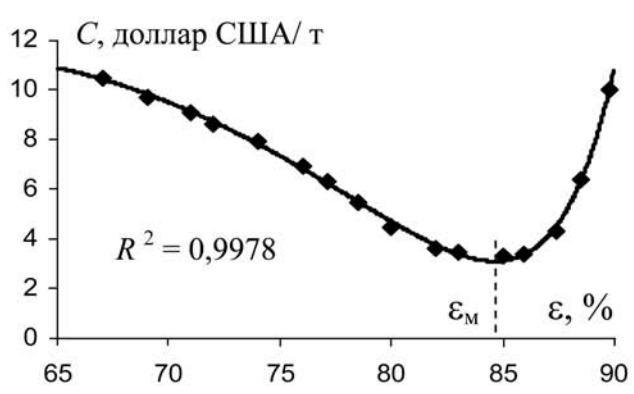
Окончание таблицы 2

Тип уравнения (самоопределяющиеся параметры)	Формула уравнения	Характерный график
Полиномные функций асимметричного распределения с самоопределяющимися смещениями x_o, основаниями a и показателями степени μ (ПАР СП), $X > 0$, $a > 1$, $\mu > 0$		
М − $X(a, \mu)$	$A[xa^{-x}]^{\mu} + B$	
М − $X(a, \mu, x > x_0)$	$A[(x - x_0)a^{x_0 - x}]^{\mu} + B$	
М − $X(a, \mu, x < x_0)$	$A[(x_0 - x)a^{x - x_0}]^{\mu} + B$	
АМ − $X(a_1, a_2, \mu_1, \mu_2, x > x_{01}, x > x_{02})$	$A_1[(x - x_{01})a_1^{x_{01} - x}]^{\mu_1} + A_2[(x - x_{02})a_2^{x_{02} - x}]^{\mu_2} + B$	
АМ − $X(a_1, a_2, \mu_1, \mu_2, x < x_{01}, x < x_{02})$	$A_1[(x_{01} - x)a_1^{x - x_{01}}]^{\mu_1} + A_2[(x_{02} - x)a_2^{x - x_{02}}]^{\mu_2} + B$	
АМ − $X(a_1, a_2, \mu_1, \mu_2, x_{01} < x < x_{02})$	$A_1[(x - x_{01})a_1^{x_{01} - x}]^{\mu_1} + A_2[(x_{02} - x)a_2^{x - x_{02}}]^{\mu_2} + B$	
Переходные квазиступенчатые с самоопределяющимися смещенным центром x_0 и коэффициентом крутизны перехода β (ПКС СП) (Г — горизонтальные, П — параллельные)		
А − $X(x_0, \beta)$Г	$\frac{A}{1 + \mathbf{e}^{\beta(X - X_0)}} + B$	
А − $X(x_0, \beta)$П	$\frac{A_1}{1 + \mathbf{e}^{\beta(X - X_0)}} + A_2(X - X_0) \times \sum_{i=2}^{i=3} \frac{1}{(-1)^i \cdot \frac{200(X - X_0)}{(X_n - X_1)}} + B$	
А − $X(x_0, \beta)$	$\frac{A_1}{1 + \mathbf{e}^{\beta(X - X_0)}} + \sum_{i=2}^{i=3} \frac{A_i(X - X_0)}{(-1)^i \cdot \frac{200(X - X_0)}{(X_n - X_1)}} + B$	
Полиномный синус (ПSin)		
А − $X(\omega, \varphi)$	$A\sin(\omega x + \varphi) + B$	
Полиномный амплитудно-зависимый синус (ПАSin)		
АМ − $X(\omega, \varphi, \beta_a)$	$A\mathbf{e}^{\beta_a x}\sin(\omega x + \varphi) + B$	
Полиномный показательный с синусом (ППSin)		
А − $X(\omega, \varphi, \beta_п)$	$A_1\sin(\omega x + \varphi) + A_2\mathbf{e}^{\beta_п x} + B$	
Полиномный показательный с амплитудно-зависимым синусом (ППАSin)		
АМ − $X(\omega, \varphi, \beta_п, \beta_a)$	$A_1\mathbf{e}^{\beta_a x}\sin(\omega x + \varphi) + A_2\mathbf{e}^{\beta_п x} + B$	

и линейно-степенной. Факторы нормального и совмещенного с ним показательного распределения выражаются в соответствующих уравнениях НР СК и ППНР СК. В шести уравнениях вида ПАР СП факторы левого или правого асимметричного распределения с одним или двумя экстремумами выражаются мультипликативным взаимодействием смещенных степенных и показательных функций. В трех видах уравнений ПКС СП факторы переходных процессов отображаются квазиступенчатой функцией с заданным или произвольным направлением ступеней. В четырех уравнениях с функцией синуса ПSin, ПАSin, ППSin, ППАSin отображено действие фактора гармонических постоянных или амплитудно-зависимых колебаний, происходящих вокруг стабильного или переменного фона. Расчет функциональных параметров в программе проводится МППВ по методикам МОЛЗТ-Ч, где Ч — число, принимающее значение от 1 до 6.

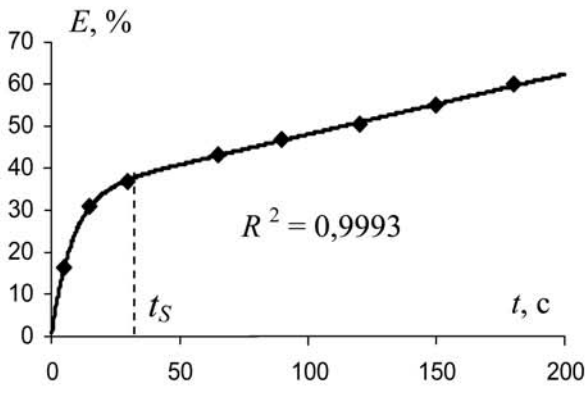
Эффективность применения программы покажем на примерах построения моделей двумерной нелинейной регрессии в некоторых экспериментальных исследованиях. Заданные узловые точки, уравнения регрессии и их графики показаны на рис. 2. На рис. 2, *а* уравнением ПС СПС, с учетом действия трех степенных факторов, и в целях управления флотационным обогащением меди, отображена зависимость ее себестоимости C от флотационного извлечения меди ε из промпродуктов. Установлено, что минимальной себестоимости соответствует значение $\varepsilon_o = 84{,}5$ %. На рис. 2, *б* зависимость сепаратного извлечения E кондиционных частиц из сырья в концентрат от времени t отображена уравнением ППЛС СК. В области перехода регрессии на асимптоту установлено время линейной стабилизации процесса $t_s = 36$ с. По уравнению определена установившаяся при этом скорость извлечения $\nu = 0{,}1433$ %/с. На рис. 2, *в* приведены график и формула уравнения ПАР СП с правосторонней асимметрией, построенные по экспериментальной выборке узловых точек. Они выражают закономерность распределения η частиц угольной пыли, образуемой в забое, по их размеру d. По уравнению определены значения размера $d_1 = 7$ мкм, $d_2 = 27$ мкм, соответствующие двум максимумам распределения. На рис. 2, *г* квазиступенчатым уравнением ПКС СП в локальной области фазового перехода позистора, в целях моделирования и управления электрической цепи установлена зависимость логарифма его электрического сопротивления от температуры. По уравнению регрессии определено, что температура логарифмического центра перехода $t_ц$ составляет 113,1 °С, а сопротивление цепи на ступенях *1* слева и *2* справа от перехода выражается соответствующими линейными уравнениями

$$\lg R = -0{,}0154t + 2{,}1505;\ \lg R = 0{,}0158t + 3{,}4608.$$



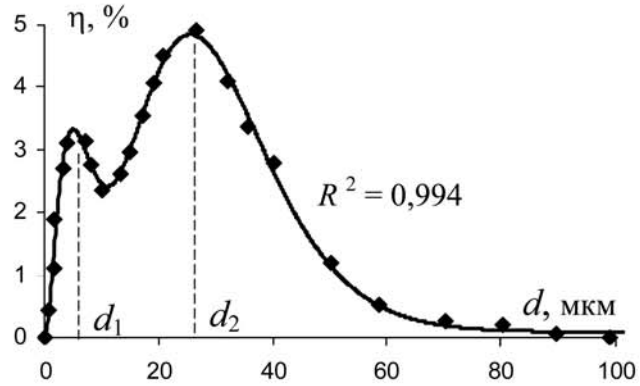
$$C = 2{,}2502\cdot10^{-59}\varepsilon^{30{,}6071} - 0{,}0914\varepsilon^{1{,}6826} + 4{,}9775\varepsilon^{0{,}8657} - 71{{,}}0971$$

а)



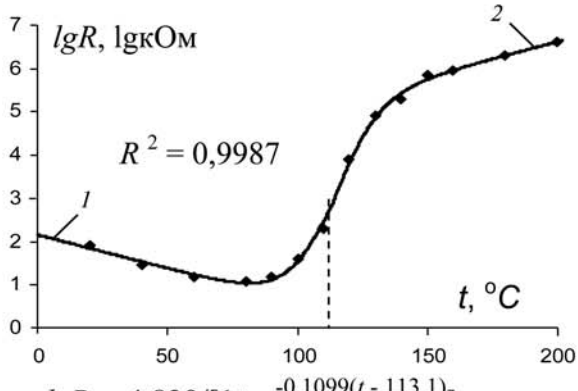
$$E = 33{,}7151(1 - \mathbf{e}^{-0{,}1253\cdot t}) + 0{,}1433\cdot t$$

б)



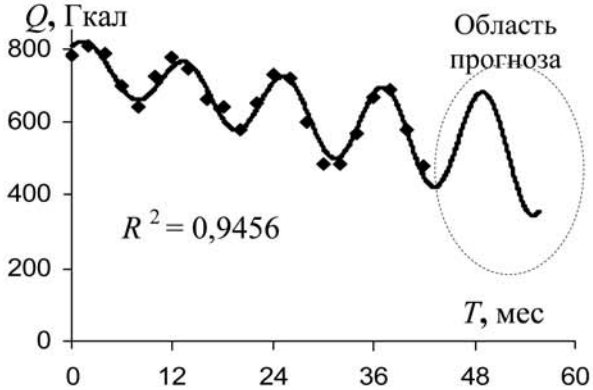
$$\eta = 1{,}0747[(d-0{,}0609)1{,}2367^{(0{,}0609-d)}]^{1{,}981} + 3{,}6706\cdot10^{-5}[(d-0{,}099)1{,}0399^{(0{,}099-d)}]^{5{,}2538} + 0{,}0894$$

в)



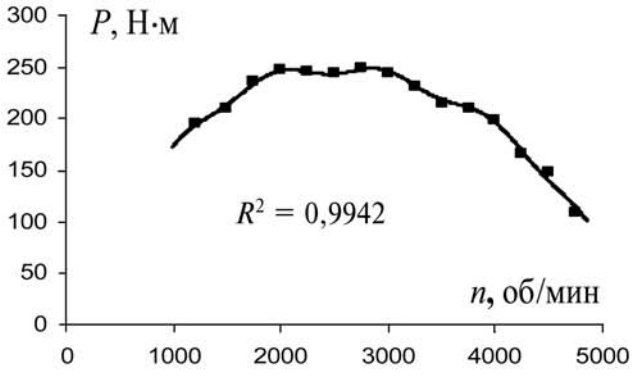
$$lgR = 4{,}839/[1+\mathbf{e}^{-0{,}1099(t-113{,}1)}] - 0{,}0154(t-113{,}1)/[1+\mathbf{e}^{1{,}111(t-113{,}1)}] + 0{,}0158(t-113{,}1)/[1+\mathbf{e}^{-1{,}111(t-113{,}1)}] + 0{,}4088$$

г)



$$Q = 220{,}09 + 551{,}4191\mathbf{e}^{-0{,}0116\cdot T} + 55{,}1744\mathbf{e}^{0{,}0203\cdot T}\sin(0{,}5309\cdot T + 0{,}7226)$$

д)



$$P = -395{,}21\mathbf{e}^{-0{,}00112\cdot n} - 6{,}506\mathbf{e}^{0{,}00072\cdot n} + 4{,}914\sin(0{,}00635\,n + 1{,}687) + 311{,}835$$

е)

Рис. 2. Примеры построения моделей двумерной функционально-факторной нелинейной регрессии программой "Тренды ФСП-1"

На рис. 2, *д* точками обозначено теплопотребление Q участка тепловой сети, обладающее свойством сезонной периодичности и накопленное по месячным T наблюдениям в течение около трех лет. Закономерность теплопотребления выражена уравнением типа ППASin, в котором учтены факторы плавного уменьшения его среднего уровня и увеличения сезонной амплитуды. Следуя приведенному уравнению, дан прогноз теплопотребления на ближайшие 6—8 месяцев. На рис. 2, *е* по ряду узловых точек экспериментальных измерений крутящего момента P силы, образуемой в автомобильном дизельном двигателе, построено комбинированное уравнение регрессии, выражающее зависимость крутящего момента от частоты вращения n вала двигателя. Сначала было построено полиномное уравнение вида ПП СК с двумя показательными функциями, отображающими влияние на регрессию фактора роста концентрации подаваемого в двигатель топлива (увеличение P в интервале n от 1200 до 2000 об/мин) и фактора ее перенасыщения (уменьшение P в интервале n от 3000 до 4800 об/мин). Однако разнознаковые отклонения узловых точек от линии такого тренда не случайны. Они указывают на существование еще одного периодически действующего фактора, связанного с особенностями конструкции и центровкой крепления двигателя. Поэтому уравнение ПП СК дополнено еще уравнением вида ПSin, построенным по данным отмеченных отклонений.

Приведем пример формирования трехмерного уравнения регрессии ФСП, построенной в целях моделирования холмистого рельефа на участке земной поверхности. На рис. 3, *а* (см. третью сторону обложки) показано распределение на данном участке узловых точек с высотой H поверхности Земли, полученное в результате геодезических измерений. Известно, что рельеф поверхности здесь умеренно гладкий, без обрывов и оврагов. Изменение высоты точек, наблюдаемое вдоль каждой горизонтальной оси координат X и Y, характеризуется монотонным подъемом, а затем спадом. Это означает, что регрессия H на данном участке должна содержать локальный максимум. Влияние осевых факторов монотонностей выразим соответствующими аддитивно взаимодействующими степенными функциями. Распределение высоты H содержит еще одну монотонность, направленную под углом к осям X и Y. Влияние данного диагонального фактора выразим мультипликативным взаимодействием степенных функций горизонтальных координат. Таким образом, первоначально уравнение регрессии зададим в следующем общем виде:

$$H = A_1 X^{\mu_1} + A_2 X^{\mu_2} + A_3 Y^{\mu_3} + A_4 Y^{\mu_4} + A_5 X^{\mu_5} Y^{\mu_6} + B.$$

Расчет оптимальных значений функциональных параметров μ_1—μ_6 проведен МППВ в последовательно сменяющихся трех ортогональных сечениях R^2_o по методике МОЛ5Т-2, МЦЛ5Т-1. Полученное в результате уравнение регрессии высоты поверхности Земли H и его график представлены на рис. 3, *б*.

Приведем пример построения четырехмерного уравнения ФСП в целях создания математической функциональной модели, описывающей распределение содержания меди в локализованной подземной части рудного блока. На этапе эксплуатационной разведки участка медно-колчеданного месторождения по результатам геологического опробования получены представленные на рис. 4, *а* (см. третью сторону обложки) данные о содержании меди C_{Cu} в ряде точек рудного блока, привязанных к трехмерному пространству по горизонтальным X, Y и вертикальным H координатам. Содержания меди в точках изображены пропорциональными по объему шарами. В их распределении вдоль каждой координаты наблюдается по одной монотонности содержания меди. Поэтому влияние на регрессию трех аддитивно взаимодействующих осевых факторов выразим соответствующими степенными функциями. Общий вид уравнения следующий:

$$C_{Cu} = A_1 X^{\mu_1} + A_2 Y^{\mu_2} + A_3 H^{\mu_3} + B.$$

Входящие в уравнение показатели степени μ_1, μ_2, μ_3 оптимизированы расчетами МППВ по методике МОЛ7Т-1 с семью опорными точками. Полученное в итоге уравнение показано на рис. 4 (см. третью сторону обложки). Графики регрессии представлены на рис. 4, *б* в виде четырех горизонтальных планов, расположенных на разных этажах по вертикали. Положение изолиний, обозначающих содержание меди в руде, показывает на планах ее объемное распределение по выделенной части рудного блока. Кроме того, по заданному бортовому содержанию меди 0,6 % оценено граничное положение в геопространстве кондиционной части рудной залежи. Границы кондиционной руды показаны пунктиром.

Список литературы

1. **Антонов В. А.** Информационная технология оптимизации новым численным методом и построения на его основе полиномных степенных трендов с самоопределяющимися показателями степени // Информационные технологии. 2009. № 8. С. 39—45.

2. **Антонов В. А.** Об одном методе построения полиномных трендов с самоопределяющимися показателями и коэффициентами // Экономика и математические методы. 2010. Т. 46, № 2. С. 78—88.

3. **Антонов В. А., Яковлев М. В**. Уравнения нелинейной регрессии, тренды двумерные функционально-факторные с самоопределяющимися параметрами и повышенной достоверностью (Тренды ФСП-1): программа для ЭВМ, РФ; регистр. номер 2011616230 / Екатеринбург: ИГД УрО РАН, 2011.

УДК 004.272.43

А. Э. Саак, канд. техн. наук, доц.,
Технологический институт
Южного федерального университета
в г. Таганроге,
e-mail: saak@tti.sfedu.ru

Полиномиальные алгоритмы диспетчеризации массивов заявок параболического типа

Рассматривается параболический тип массива заявок пользователей на компьютерное обслуживание в Grid-системах, многопроцессорных вычислительных системах. Предлагаются и исследуются начально-уровневый и возвратный центрально-уровневый полиномиальные алгоритмы назначения заявок параболического квадратичного типа и даются рекомендации о возможности использования в диспетчере как МВС, так и центра Grid-технологий.

Ключевые слова: *Grid-система, многопроцессорная вычислительная система, диспетчирование, параболический квадратичный тип массива требований пользователей, начально-уровневый полиномиальный алгоритм, возвратный центрально-уровневый полиномиальный алгоритм*

Введение

В работах [1—3] для массивов заявок пользователей на компьютерное обслуживание в Grid-системах, многопроцессорных вычислительных системах (МВС) [6—11] определена квадратичная типизация. В [12—15] исследовалась оптимальная укладка последовательности квадратов, являющейся модельным примером массива кругового типа. В [16] рассматривалась оптимальная укладка последовательности прямоугольников с постоянным периметром, являющейся модельным примером массива гиперболического типа. В [4] для требований кругового квадратичного типа, а в [5] для требований гиперболического квадратичного типа предложены и исследованы полиномиальные алгоритмы распределения ресурсов, адаптированные под соответствующий тип массива заявок пользователей. В [5] приведена оценка качества полиномиальных алгоритмов посредством эвристической меры ресурсной оболочки. В настоящей статье предлагаются и исследуются полиномиальные алгоритмы диспетчеризации, учитывающие свойства параболического типа массива заявок — быстрое убывание высот прямоугольников при геометрическом представлении требований.

Начально-уровневый алгоритм диспетчеризации массивом заявок параболического типа

При представлении заявки пользователя на обслуживание диспетчером центра Grid-технологий или операционной системы МВС координатным ресурсным прямоугольником горизонтальное и вертикальное измерения принимаются равными соответственно числу единиц ресурса процессоров и времени, требуемому для обработки. Символом $a(j_1) \times b(j_1)$ или $[a(j_1), b(j_1)]$ обозначается j_1-я заявка, требующая $a(j_1)$ единиц процессоров и $b(j_1)$ единиц времени.

Приведем и исследуем начально-уровневый алгоритм диспетчеризации линейными параболическими полиэдралями координатных ресурсных прямоугольников $\bigcup_{j_1=0}^{k-1} [a(j_1), b(j_1)]$.

За уровень принимаем высоту начального ресурсного прямоугольника $b(0)$ и получаем начальную ресурсную оболочку (рис. 1). На втором шаге вдоль линии $Y_1 = a(0)$ вертикально суперпозируются ресурсные прямоугольники $\bigcup_{j_1=1}^{q_1} [a(j_1), b(j_1)]$ до наилучшего приближения уровня с недостатком $\sum_{j_1=1}^{q_1} b(j_1) = b(0) - 0$, где q_1 — мощность ресурсных прямоугольников, суперпозированных во втором слое. Строится ресурсная оболочка полученной аддитивной графики (рис. 2).

На третьем шаге вдоль правой стороны достигнутой ресурсной оболочки $Y_1 = a(0) + \max\limits_{1 \leqslant j_1 \leqslant q_1} a(j_1)$ вертикально суперпозируются последующие ресурсные прямоугольники $\bigcup_{j_1=q_1+1}^{q_2} [a(j_1), b(j_1)]$ до наилучшего приближения уровня с недостатком $\sum_{j_1=q_1+1}^{q_2} b(j_1) = b(0) - 0$, где q_2 — мощность ресурсных прямоугольников, суперпозированных в третьем слое. Строится новая ресурсная оболочка (рис. 3).

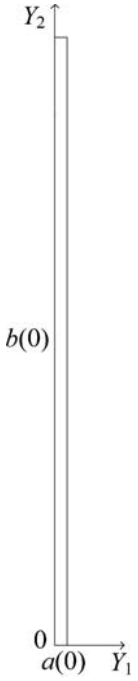


Рис. 1. Начальная ресурсная оболочка

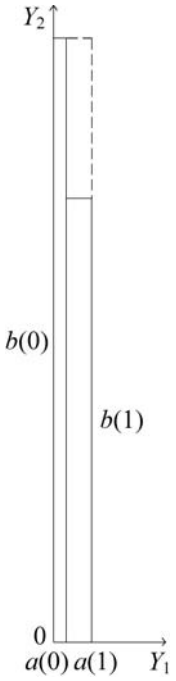


Рис. 2. Первая ресурсная оболочка

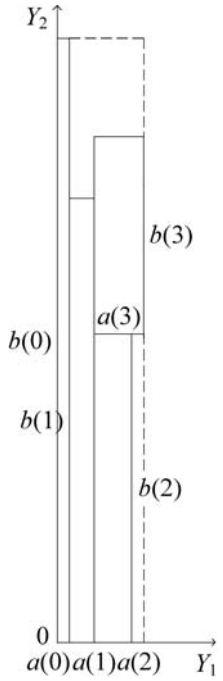


Рис. 3. Вторая ресурсная оболочка

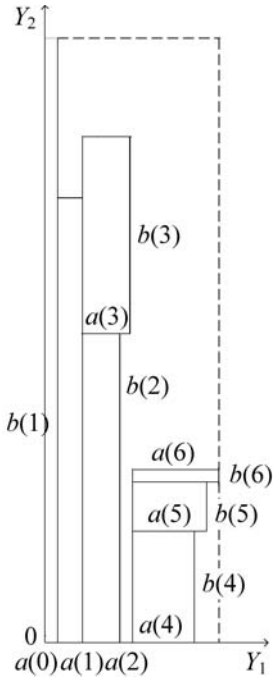


Рис. 4. Конечная ресурсная оболочка

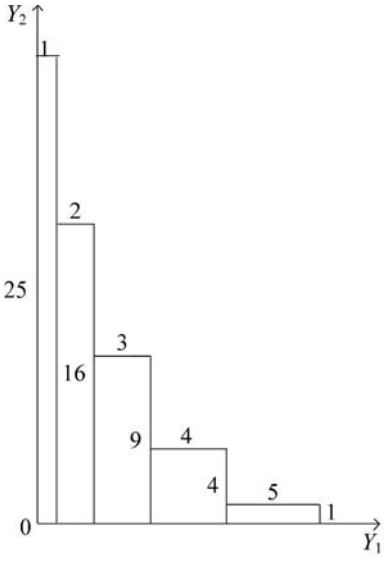


Рис. 5. Параболический массив $j_1 \times (k - j_1)^2$, $k = 6$

На следующем шаге вдоль правой стороны достигнутой ресурсной оболочки $Y_1 = a(0) + \max\limits_{1 \leqslant j_1 \leqslant q_1} a(j_1) + \max\limits_{q_1 + 1 \leqslant j_1 \leqslant q_2} a(j_1)$ вертикально суперпозируются последующие ресурсные прямоугольники $\bigcup\limits_{j_1 = q_2 + 1}^{q_3} [a(j_1),\ b(j_1)]$ до наилучшего приближения уровня с недостатком $\sum\limits_{j_1 = q_2 + 1}^{q_3} b(j_1) = b(0) - 0$, где q_3 — мощность ресурсных прямоугольников, суперпозированных в четвертом слое. Строится конечная ресурсная оболочка (рис. 4).

Введенный таким образом начально-уровневый алгоритм повторяем до полного исчерпания ресурсных прямоугольников массива. Качество алгоритма оцениваем посредством эвристической меры ресурсной оболочки, учитывающей как меру (площадь), так и форму (асимметрию) оболочки и суммарную меру охватываемых ресурсных прямоугольников массива [5].

В качестве примера метода начально-уровневой параболической локализации выбираем массив натуральных параболических ресурсных прямоугольников $j_1 \times (k - j_1)^2$. При $k = 6$ (рис. 5) за уровень принимаем высоту начального ресурсного прямоугольника $b(0) = 25$ и получаем начальную ресурсную оболочку (рис. 6). Второй (рис. 7) и третий (рис. 8) шаги начально-уровневого алгоритма диспетчеризации параболической линейной полиэдрали

Таблица 1

Эвристические меры ресурсных оболочек начально-уровневого алгоритма

k	Эвристическая мера	k	Эвристическая мера
6	2,29	11	3,45
7	2,69	12	3,63
8	2,84	13	3,70
9	3,11	14	3,85
10	3,21	15	3,82

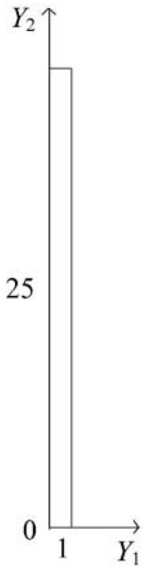


Рис. 6. Начальная ресурсная оболочка

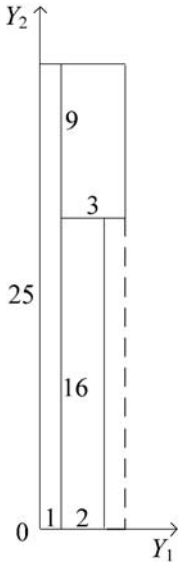


Рис. 7. Первая ресурсная оболочка

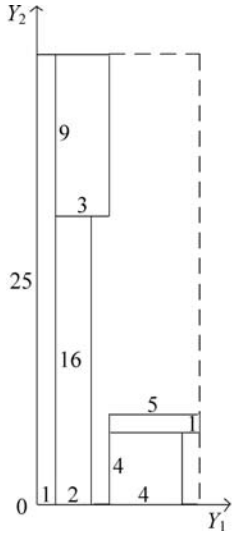


Рис. 8. Конечная ресурсная оболочка

приводят к ресурсной оболочке с эвристической мерой $\frac{1}{2}\left[\frac{9 \times 25}{105} + \frac{(25-9)^2}{105}\right] = 2{,}29$.

Приведем в табл. 1 эвристические меры ресурсных оболочек начально-уровневого алгоритма для последовательности натуральных параболических ресурсных прямоугольников $1 \times (k-1)^2$, $2 \times (k-2)^2$, ..., $(k-1) \times 1$.

Возвратный центрально-уровневый алгоритм диспетчеризации массивом заявок параболического типа

Для линейной параболической полиэдрали $\bigcup_{j_1=0}^{k-1} [a(j_1),\ b(j_1)]$ находим средний уровень $\overline{b} = \frac{1}{k}\sum_{j_1=0}^{k-1} b(j_1)$ и центральную грань j_1^*, ближайшую к точке достижения среднего уровня $b(j_1^*) = \overline{b} \pm 0$.

Грани $j_1 \leqslant j_1^*$ относим в левый блок, $j_1 > j_1^*$ — в правый блок. К последним применяем уровневый алгоритм для уровня $\overline{b}$. А именно, грани правее центра суперпозируем вертикально над гранью $[a(j_1^* + 1), b(j_1^* + 1)]$ до наилучшего приближения к уровню $\overline{b}$ суммарными высотами вертикальной полиэдрали

$$\sum_{j_1 = j_1^* + 1}^{q} b(j_1) = \overline{b} - 0.$$

Из оставшихся граней низкого уровня $b(j_1) < \overline{b}$ формируем последующие вертикальные полиэдрали суммарной высоты $\overline{b} - 0$, подвергая последние динамической суперпозиции к предыдущим вдоль горизонтальной координаты. В итоге получаем ресурсную оболочку правого блока $[(A, B)]$.

В возвратном центрально-уровневом алгоритме к левой стороне полученной оболочки суперпозируем начальную грань $[a(0), b(0)]$ максимальной высоты, получая угловую область начального полиэдра (рис. 9).

В угловой области данного полиэдра вершинно-диагонально синтезируем линейную полиэдраль левого блока за вычетом начальной грани. Вычисляем эвристическую меру получаемой ресурсной оболочки (рис. 10) с измерениями $\max\{a(0) + A,\ A^*\}$ — по горизонтали, $\max\{b(0),\ b(1) + B\}$ — по вертикали, где $A^* = \sum_{j_1=0}^{j_1^*} a(j_1)$.

Возвращаем вершинно-диагональный блок в первоначальное положение. Синтезируем с оболочкой правого блока $[(A, B)]$ начальную $[a(0), b(0)]$ и первую

Рис. 9. Суперпозиция начальной грани и оболочки правого блока

$[a(1),\ b(1)]$ грани и подвергаем вершинно-диагональному синтезу в новую угловую область упомянутую линейную полиэдраль левого блока без начальной и первой граней. Вычисляем эвристическую меру получаемой ресурсной оболочки (рис. 11) с измерениями $\max\{a(0) + a(1) + A,\ A^*\}$ — по горизонтали, $\max\{b(0),\ b(j_1^*) + B\}$ — по вертикали. На последнем шаге осуществляем горизонтальный синтез с оболочкой правого блока $[(A, B)]$ линейной полиэдрали левого блока $\bigcup_{j_1=0}^{j_1^*} [a(j_1),\ b(j_1)]$ и вычисляем эвристическую меру получаемой ресурсной оболочки (рис. 12) с измерениями $(A^* + A)$ — по горизонтали, $\max\{b(0),\ B\}$ — по вертикали.

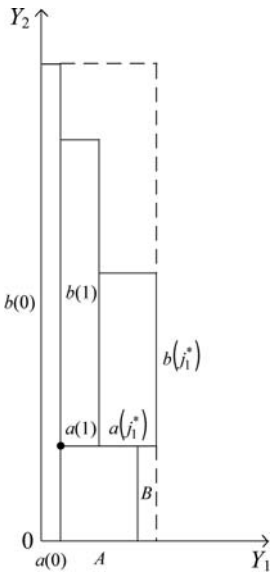


Рис. 10. Ресурсная оболочка первого шага алгоритма

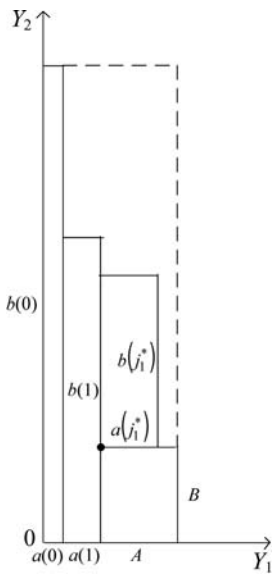


Рис. 11. Ресурсная оболочка второго шага алгоритма

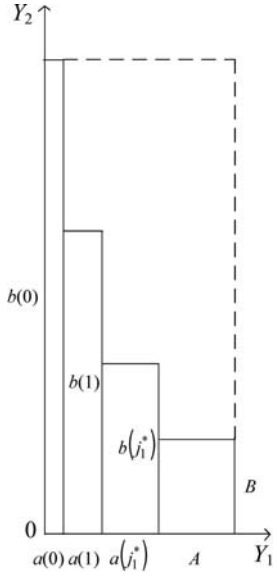


Рис. 12. Ресурсная оболочка конечного шага возвратного алгоритма

Повторение возвратных перераспределений граней параболической полиэдрали продолжаем до получения эвристики $\frac{1}{2} + 0$ или до исчерпания шагов алгоритма. Вариант с меньшей эвристической мерой считаем наилучшим.

В качестве примера метода центрально-уровневой параболической локализации возьмем линейную полиэдраль натуральных параболических ресурсных прямоугольников $j_1 \times (k - j_1)^2$, $k = 6$ предыдущего параграфа.

Имеем среднюю высоту $\overline{b} = 11$. Центральная грань $j_1^* = 3$. Оболочка граней $j_1 > j_1^*$ правого блока приведена на рис. 13.

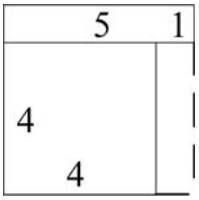


Рис. 13. Ресурсная оболочка граней правого блока

Первую грань 1×25 суперпозируем с левой стороны правого блока и вершинно-диагонально синтезируем грани с $j_1 = 2$, $j_1 = 3$ в угловую область (рис. 14). Вычисляем меру эвристики

$$\frac{1}{2}\left[\frac{6 \times 25}{105} + \frac{(25-6)^2}{105}\right] = 2{,}43.$$

Возвращаем вершинно-диагональный блок (грани 2×16; 3×9) в первоначальное положение. Синтезируем с оболочкой правого блока первую и вторую грани 1×25; 2×16. Вновь строим ресурсную оболочку (рис. 15) и вычисляем меру эвристики

$$\frac{1}{2}\left[\frac{8 \times 25}{105} + \frac{(25-8)^2}{105}\right] = 2{,}33.$$

Возвращаем вершинно-диагональный блок (грань 3×9) в первоначальное положение. Синтезируем с оболочкой правого блока первую, вторую и третью грани 1×25; 2×16; 3×9. Вновь строим ресурсную оболочку (рис. 16) и вычисляем меру эвристики

$$\frac{1}{2}\left[\frac{11 \times 25}{105} + \frac{(25-11)^2}{105}\right] = 2{,}24.$$

Видим, что более предпочтительным является третий шаг центрально-уровневого алгоритма с меньшей эвристической мерой 2,24.

Приведем в табл. 2 эвристические меры ресурсных оболочек центрально-уровневого алгоритма для последовательности натуральных параболических ресурсных прямоугольников $1 \times (k-1)^2$, $2 \times (k-2)^2$, ..., $(k-1) \times 1$.

Графики эвристической меры ресурсных оболочек начально-уровневого и центрально-уровневого полиномиальных алгоритмов диспетчеризации линейными полиэдралями $1 \times (k-1)^2$, $2 \times (k-2)^2$, ..., $(k-1) \times 1$ параболического типа показаны на рис. 17.

Видим, что более предпочтительным является центрально-уровневый алгоритм диспетчирования, как имеющий меньшую эвристическую меру ресурсных оболочек. Сравнение эвристических мер ресурсных оболочек подтверждает целесообразность использования предложенного центрально-уровневого алгоритма при диспетчеризации линейными полиэдралями параболического типа.

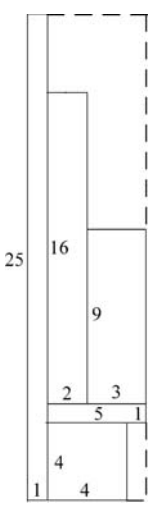


Рис. 14. Ресурсная оболочка первого шага

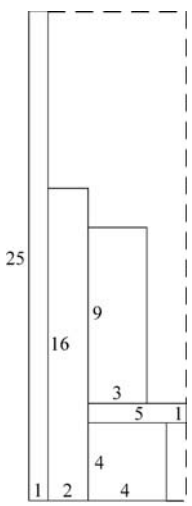


Рис. 15. Ресурсная оболочка второго шага

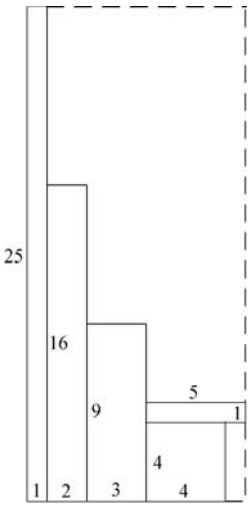


Рис. 16. Ресурсная оболочка третьего шага

Таблица 2

Эвристические меры ресурсных оболочек центрально-уровневого алгоритма

k	Эвристическая мера	k	Эвристическая мера
6	2,24	11	3,25
7	2,57	12	3,26
8	2,76	13	3,50
9	2,90	14	3,55
10	3,15	15	3,60

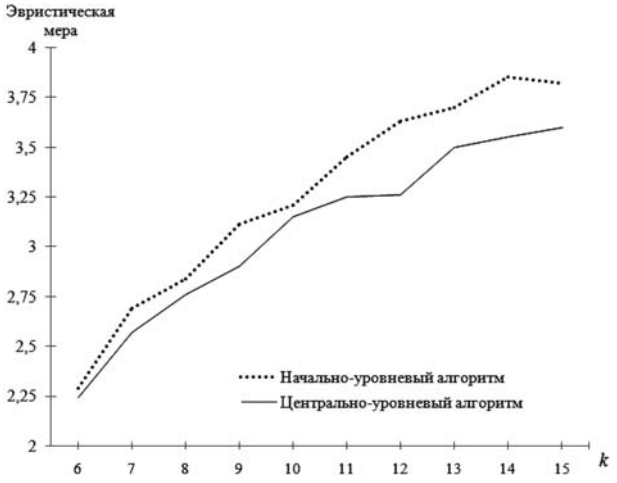


Рис. 17. Эвристические меры ресурсных оболочек начально-уровневого и центрально-уровневого алгоритмов

Заключение

Для параболического типа массива заявок пользователей рассмотрены полиномиальные алгоритмы диспетчеризации, учитывающие свойства указанного квадратичного типа. Возвратный алгоритм может быть предложен также для круговых и гиперболических типов линейных полиэдралей с варьированием среднего измерения согласно имеющемуся типу массива требований. Приведены фундаментальные примеры распределения вычислительных ресурсов и даны рекомендации о возможности использования разработанных алгоритмов в диспетчере как МВС, так и центра Grid-технологий.

Список литературы

1. **Саак А. Э.** Локально-оптимальные ресурсные распределения // Информационные технологии. 2011. № 2. С. 28—34.
2. **Саак А. Э.** Алгоритмы диспетчеризации в Grid-системах на основе квадратичной типизации массивов заявок // Информационные технологии. 2011. № 11. С. 9—13.
3. **Саак А. Э.** Диспетчеризация в Grid-системах на основе однородной квадратичной типизации массивов заявок пользователей // Информационные технологии. 2012. № 4. С. 32—36.
4. **Саак А. Э.** Сравнительный анализ полиномиальных алгоритмов диспетчеризации в Grid-системах // Информационные технологии. 2012. № 9. С. 28—32.
5. **Саак А. Э.** Полиномиальные алгоритмы диспетчеризации массивов заявок гиперболического типа // Информационные технологии. 2013. № 3. С. 33—36.
6. **Барский А. Б.** Параллельные информационные технологии. М.: ИНТУИТ; БИНОМ. Лаборатория знаний, 2007. 503 с.
7. **Барский А. Б.** Алгоритмические, архитектурные и структурные методы организации управляющих процессов в виртуальном пространстве средств Grid-системы // Информационные технологии. 2012. № 5. С. 2—6.
8. **Васенин В. А., Шундеев А. С.** Эволюция технологии Grid // Информационные технологии. 2012. № 1. С. 2—9.
9. **Хорошевский В. Г.** Архитектура вычислительных систем. М.: Изд-во МГТУ им. Н. Э. Баумана, 2005. 512 с.
10. **Воеводин В. В., Воеводин Вл. В.** Параллельные вычисления. СПб.: БХВ-Петербург, 2002. 608 с.
11. **Каляев И. А., Левин И. И., Семерников Е. А., Шмойлов В. И.** Реконфигурируемые мультиконвейерные вычислительные структуры. Изд. 2-е, перераб. и доп. / Под общ. ред. И. А. Каляева. Ростов н/Д: Издательство ЮНЦ РАН, 2009. 344 с.
12. **Korf R.** Optimal rectangle packing: Initial results // Proc. of the thirteenth international conference on automated planning and scheduling (ICAPS 2003) Trento, Italy, June 9—13, 2003. P. 287—295.
13. **Korf R.** Optimal rectangle packing: New results // Proc. of the fourteenth international conference on automated planning and scheduling (ICAPS 2004) Whistler, British Columbia, Canada, June 3—7, 2004. P. 142—149.
14. **Korf R., Moffitt M., Pollack M.** Optimal rectangle packing. Annals of Operations Research. Number 1. 2010. Vol. 179. P. 261—295.
15. **Korf R., Huang E.** New Improvements in Optimal Rectangle Packing // Proc. of the 21st International Joint Conference on Artificial Intelligence (IJCAI 2009) Pasadena, California, USA, July 11—17, 2009. P. 511—516.
16. **Korf R., Huang E.** (2010). Optimal Rectangle Packing on Non-Square Benchmarks // Proceedings of the twenty-fours AAAI Conference on Artificial Intelligence (AAAI-10). Atlanta, Georgia, USA, July 11—15, 2010. P. 83—88.

УДК 004.75

И. П. Карпова, канд. техн. наук, доц.,
e-mail: karpova_ip@mail.ru
МИЭМ НИУ ВШЭ

Хранение данных системой автономных устройств

Рассматривается вопрос организации хранения данных в памяти системы автономных устройств, оснащенных датчиками и собирающими информацию о состоянии окружающей среды. Предлагается метод организации записи данных в память, который обеспечит рациональное использование памяти в условиях ограничений на ресурсы.

Ключевые слова: *система сбора и обработки данных, хранение данных, кольцевой буфер*

Введение

В настоящее время активно развиваются беспроводные сети сбора и обработки данных. В качестве примеров таких сетей можно привести беспроводные сенсорные сети (БСС) [1] или системы автономных мобильных роботов [2]. В узлах сети располагаются устройства (мобильные или стационарные), оснащенные датчиками. Номенклатура датчиков зависит от той задачи, которая стоит перед сетью. Обычно такие сети предназначены для мониторинга состояния окружающей среды, в системах безопасности и пр. [3].

Показания датчиков сохраняются в памяти устройства для дальнейшей обработки и/или передачи данных на центральный компьютер. Центральный компьютер может запрашивать с узлов информацию о показаниях датчиков в определенный момент времени и в определенной точке пространства. В целом система должна уметь отвечать на вопросы, включающие пространственно-временные характеристики, например: "Что наблюдалось в тот или иной момент времени и в окрестности определенной точки пространства?" Таким образом, для получения целостной картины устройства должны иметь единую привязку по времени и координатам пространства и должны уметь определять свое местоположение (относительно маяков, меток или других ориентиров).

Работа сенсорной сети построена на передаче данных от одного узла другому, причем на передачу данных тратится значительно больше энергии, чем на сбор или обработку этих данных. В случае передачи собранных данных на центральный узел сразу

после опроса датчиков устройство может не иметь собственной памяти для хранения данных. Но здесь возникают ограничения, связанные с энергопотреблением [1, 3]. Для сенсорных сетей проблема энергосбережения стоит очень остро, так как сенсорные устройства работают автономно, а возможность замены элементов питания в них не предусмотрена. В системах мобильных роботов другая проблема: при перемещении робот может выйти из зоны приема центрального компьютера и не сможет передать данные в момент их получения с датчиков.

Вместе с тем, далеко не все задачи, решаемые с помощью подобных систем, требуют немедленной передачи собранных данных. В качестве примера можно привести системы, центральный компьютер которых опрашивает устройства периодически (например, в целях мониторинга загрязнения окружающей среды) или при возникновении какого-то нештатного события (аварии на химическом производстве и т. п.). Более того, общая тенденция в развитии подобных систем такова: по возможности переложить обработку данных на сами узлы, т. е. передавать на центральный узел не "сырые" данные, а результаты их обработки. В этом случае инициатива о передаче данных вообще может возлагаться на устройство в узле. Устройство обрабатывает данные за некоторый период времени, определяет тенденцию развития ситуации или получает какую-то статистику, после чего передает результаты на центральный узел. Это уменьшает трафик в сети и сокращает ее энергопотребление.

Во всех рассмотренных ситуациях устройствам необходимо хранить собранные данные. Причем чем больше будет тот период, в течение которого данные будут храниться, тем выше будет ценность собранной информации: можно качественнее восстановить предшествующую событию ситуацию и точнее сделать прогноз на будущее.

Таким образом, возникает задача разработки такого способа управления собранными данными, который обеспечивал бы к ним доступ без потребности массовой передачи данных в центральные узлы [4].

Постановка задачи

Пусть имеется несколько автономных устройств (мобильных или стационарных), оснащенных определенным числом датчиков. С помощью этих датчиков они могут собирать информацию об окружающем мире, например, о температуре, освещенности, химическом составе воздуха и т. п. Условия работы системы таковы:

- каждое устройство оснащено произвольным набором датчиков;
- все устройства работают асинхронно, но для системы в целом поддерживается единая служба времени;
- время отсчитывается тактами, размер такта для всех устройств одинаков (например, 1 с);
- устройство снимает показания с датчиков с периодичностью, которая определяется его программой и не зависит от других устройств;
- в каждый момент времени устройство может снимать показания с произвольного числа своих сенсоров;
- данные необходимо сохранять в памяти устройства для дальнейшей обработки и/или передачи по заранее обусловленному адресу;
- все показания снимаются с определенной погрешностью, для каждого датчика (показателя) значение этой погрешности свое. Оно зависит от качества того датчика, которым оснащено устройство, условий работы и пр.;
- нет никаких гарантий, что каждый датчик ответит на запрос: на датчике может произойти сбой, и тогда данные от него не поступят;
- объем памяти устройства невелик; при исчерпании этого объема запись должна проводиться поверх самых старых по времени показаний;
- важны не просто последние по времени значения данных, а их изменения. Если данные не изменились, то имеет смысл хранить период, в течение которого значение показателя не менялось. Это позволит в том же объеме памяти сохранить данные за более длительный промежуток времени.

Определим круг задач, которые стоят перед такой системой:

- хранение данных в базе данных (БД) устройства. Показания датчиков привязаны ко времени и местоположению устройства;
- обеспечение ответов на запросы о значениях показателей в окрестностях определенного момента времени и в окрестностях определенной точки пространства.

В данной статье будет рассмотрен вопрос организации хранения данных в такой системе в условиях ограниченного объема памяти каждого устройства.

Организация хранения данных

Для начала требуется определить, в каком виде наиболее целесообразно хранить эти данные.

В нашем случае все данные можно разделить на две группы: справочные данные и показания датчиков. К первой группе относятся:

- настройки (интервал времени для определения временной окрестности; диапазон координат для определения окрестностей точки пространства и т. д.);
- перечень показателей, которые можно снять с датчиков (наименование показателя, погрешность измерения показателя и пр.).

Пространственно-временные настройки должны быть одинаковы для всех устройств, а перечень показателей и погрешности их измерений могут быть разными, так как они зависят от конкретной номенклатуры датчиков. Интервал времени для определения временной окрестности относительно

момента t_0 может быть несимметричным, например, $[t_0 - 1, t_0 + 5]$. А в качестве диапазона координат для определения окрестностей точки пространства имеет смысл взять радиус окрестности точки x_0, $O(x_0, r)$.

Что же касается хранения показаний датчиков, можно хранить данные в виде реляционной таблицы (или таблиц). Это хорошо известный механизм, применение которого позволило бы при передаче данных на центральную машину без проблем загружать эти данные в реляционную базу данных и использовать для обработки данных тот же язык SQL.

Но при ограничениях по объему памяти использование для хранения данных реляционной модели является не самым удачным решением. Продемонстрируем это на конкретном примере. В реляционной модели возможны два варианта хранения данных о показаниях датчиков:

- в виде кортежей: время плюс набор всех показателей;
- в виде троек (время, номер показателя, значение показателя).

Рассмотрим две крайние ситуации:

1) в каждый момент времени происходит съем показателей со всех датчиков, и значения всех показателей меняются;

2) в каждый момент времени изменяется только один показатель или происходит опрос только одного датчика.

В первом случае оптимальным является хранение данных кортежами, так как при этом никакая информация не теряется, а время хранится только один раз на кортеж. Во втором случае целесообразно хранить данные тройками (время, номер, значение), чтобы не хранить "пустоту". В реальности ни первой, ни второй ситуации в чистом виде не будет. В зависимости от условий, в которых будет функционировать устройство, и от задач, которые он будет решать, ситуация будет ближе либо к первому, либо ко второму варианту. Но и в том и в другом случае память будет расходоваться неэкономно.

Приведем несложные расчеты для случая, когда значения всех показателей занимают одинаковое количество памяти (4 байт). 4 байта для хранения времени (числа тактов после начала работы) — это свыше ста лет при размере такта 1 с, т. е. более чем достаточно. Пусть N — это число датчиков на устройстве. При хранении данных кортежами на каждый кортеж уйдет $(N + 1)4$ байт. Обычно число датчиков на устройстве от 5 до 10. Таким образом, пропуск даже одного значения дает потери примерно 10...15 %.

Вместе с тем, если хранить данные тройками, то каждая из них будет занимать 9 байт: время (4 байт) + + номер показателя (1 байт) + значение (4 байт). Такая схема хранения имеет смысл только в той ситуации, при которой в каждый момент времени собирается не более 45 % от общего числа показателей:

$$9M < 4(N + 1);$$

$$\frac{M}{N+1} < 0{,}44,$$

где M — это число реально снятых показаний. При этом M может меняться динамически от 1 до N в зависимости от реальной ситуации. И если брать за основу реляционную модель (хранить данные в виде реляционных таблиц), то либо память будет расходоваться неэффективно, любо придется усложнять программу и динамически менять способ хранения данных, записывая их в разные таблицы. Такой подход имеет и еще один недостаток: усложнение реализации запросов к данным из-за того, что они хранятся в двух таблицах разной структуры.

Исходя из вышесказанного предлагается другой вариант для организации хранения значений показателей: хранить данные кортежами, но включать в кортеж только реальные значения. Это можно организовать одним из двух способов:

1. Если значение каждого показателя имеет фиксированную длину, то эта длина хранится в таблице настроек. А каждому кортежу показаний датчиков приписывается в начало битовая строка признаков, в которой i-й бит соответствует i-му датчику:

1 — есть значение,

0 — нет значения.

2. Если значения показателей могут занимать разный объем памяти (например, фотографии разной четкости или запись звука разной продолжительности), то размер значения хранится в самом кортеже непосредственно перед значением показателя.

Такой подход позволит хранить только реально полученные с датчиков показания.

Примечание: в реальной ситуации возможны и промежуточные варианты, когда часть показаний имеет фиксированную длину и хранится в начале кортежа (первый способ), а часть показаний — переменную длину (второй способ).

Важно также различать ситуации, когда показания не были сняты (по причине сбоя датчика, например) и когда показания не записывались в память, потому что не изменились. Для отражения этих фактов необходима еще одна битовая строка, разряды которой будут интерпретироваться следующим образом: 1 — показания не сняты, 0 — показания не изменились.

Длина каждой из этих битовых строк при указанном диапазоне возможных значений N составит 2 байт. Таким образом, для показателей фиксированной длины каждый кортеж займет $(8 + 4N)$ байт. На рис. 1 приведен пример такой организации для $N = 6$ и временно́й отметки $t = 29$. Первая битовая строка означает, что из шести показателей хранятся

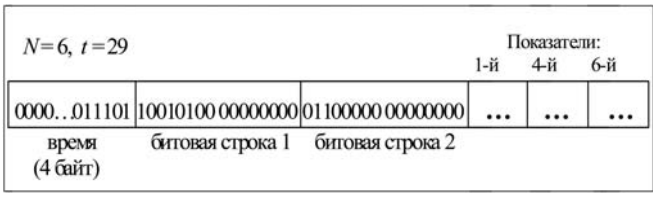


Рис. 1. Пример кортежа данных

три (с номерами 1, 4, 6), при этом показатели 2—3 не сняты, а показатель 5 не изменился.

При обработке данных во второй битовой строке будут анализироваться только те биты, для которых в первой битовой строке соответствующий бит установлен в 0.

Организация перезаписи данных

При поступлении данных (кортежей) они записываются в память до тех пор, пока ее объем не исчерпан. При этом записываются только изменившиеся значения показателей. После исчерпания объема памяти необходимо осуществлять запись новых кортежей поверх самых старых (устаревших, неактуальных) данных. То есть память должна быть организована в виде *кольцевого буфера*, который имеет два указателя: на первый и на последний кортеж, записанный в память.

Здесь возникают две сложности:

1) как определить, изменилось ли показание датчика по сравнению с предыдущим значением?

2) как определить, можно ли перезаписывать новый кортеж поверх старого, или в нем хранится неизменное значение показателя и его нельзя потерять?

Кэш. Для ответа на первый вопрос предлагается выделить специальную область памяти (*кэш*), в котором будут храниться последние значения показаний датчиков (без привязки ко времени). После очередного опроса датчиков каждое полученное значение показателя сравнивается с тем, которое хранится в кэше. Если разность между новым значением показателя и тем, которое хранится в кэше, превышает погрешность, то это значение записывается в кэш и вносится в кортеж значений. Если же эта разность меньше погрешности для данного показателя, т. е. значение не изменилось, то оно не вносится в кортеж значений (причем i-е биты первой и второй битовых строк устанавливаются в 0). В кэше при этом также остается прежнее значение. Это необходимо для того, чтобы предотвратить "дрейф" значения показателя, так как отношение "примерно равно" не является транзитивным.

Теперь рассмотрим вопрос с актуальностью данных. Как определить, является ли перезаписываемое значение определенного показателя устаревшим или это неизменившееся значение? Сравнение с тем значением, которое хранится в кэше, ничего не даст. Перезаписываемое значение может быть равным тому, которое хранится в кэше, но это не значит, что оно не менялось за период между записью кортежа в память и получением того значения, которое хранится в кэше. Конечно, можно просмотреть всю память в поисках значения данного показателя с более поздней временно́й отметкой и, если такого нет, не перезаписывать значение данного показателя. Но это потребует много времени, потому что кортежи в общем случае имеют разную длину, и их просмотр можно осуществить только последовательно.

Для решения этой задачи предлагается такой подход. Если в новом кортеже отсутствует значение определенного показателя, то можно считать, что это значение не изменяется и перезаписывать его нельзя. Но вместе с ним останется весь кортеж. Хранение таких кортежей — побочный эффект: будут храниться устаревшие показатели датчиков. Это серьезный недостаток: эти данные уже неактуальны, но они занимают память. Потери памяти в крайнем случае могут составлять до $(N-1)$-й записи, причем устаревшими в каждой записи будет $(N-1)$ показание:

$$4(N-1)(N-1) = 4(N-1)^2$$

при условии, что каждый показатель занимает 4 байт и $N-1$ показателей не меняют своего значения, но фиксация неизменного значения каждого показателя произошла в разные моменты времени.

Кроме того, периоды неизменности значений разных показателей в общем случае не совпадают друг с другом, и это может привести к искажению сведений о значениях изменяющихся во времени показателей: какие-то из них мы будем хранить (вместе с неизменными), а какие-то — удалять. Поэтому предлагается не хранить эти устаревшие значения, а "упаковывать" кортежи, оставляя только неизменные значения и время, когда это значение было получено (естественно, изменяя соответствующим образом битовые строки). Таких записей будет немного — не более $(N-1)$. Они сохранят структуру обычного кортежа, но будут хранить только одно значение.

К сожалению, если оставить эти упакованные кортежи в основной памяти, то могут возникнуть другие проблемы. Упаковав кортеж с временной отметкой t_k и значением i-го показателя $P[i]$, мы можем в следующий момент времени t_{k+n} получить кортеж, в котором i-й показатель будет присутствовать, и перезапишем кортеж с временно́й отметкой t_{k+1}. Таким образом, возникнет искажение данных: в соответствии с принятыми правилами такие данные будут означать, что в диапазоне $[t_k; t_{k+2}]$ значение $P[i]$ не изменялось, что не соответствует действительности. Пример такого искажения приведен на рис. 2. Из него получается, что в период $t_k \ldots t_{k+2}$ значение $P[2]$ оставалось неизменным (и было равно 6), а это не так: $P[2]$ $(t_{k+1}) = 5$,

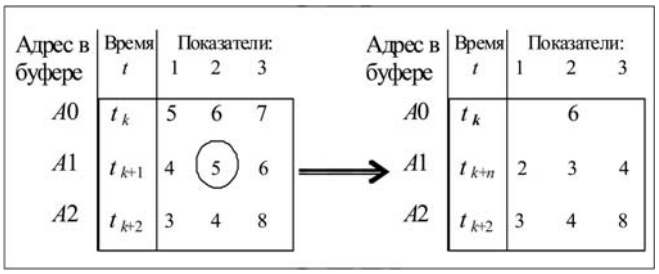


Рис. 2. Пример искажения данных

но это значение потеряно при записи кортежа (2, 3, 4) с временно́й отметкой t_n поверх кортежа с временно́й отметкой t_{k+1}.

Буфер. Для того чтобы избежать таких искажений, эти упакованные кортежи необходимо хранить в отдельной области памяти. Назовем эту область *буфером*. Правила работы с буфером будут следующими. Если в новом кортеже, который необходимо перезаписать поверх старого кортежа, отсутствует *i*-е значение, а в старом кортеже оно есть, то оно записывается в буфер на место *i*-го показателя вместе со своей временно́й отметкой, после чего новый кортеж записывается поверх старого. (Естественно, если в новом кортеже отсутствуют несколько показаний, все они из старого кортежа попадают в буфер.) Число ячеек в буфере равно числу показателей.

Начинать запись в буфер значения *i*-го показателя надо только тогда, когда значения этого показателя перестали появляться в новых данных. Если при пропуске значений помещать *i*-й показатель в буфер, а при появлении новых значений — очищать буфер, то может возникнуть ситуация, когда будет доступно только одно значение. Это может произойти, если в цикле перезаписи это значение появляется только один раз. Естественно, это недопустимо.

Вместе с тем, если при наличии значения $P[i]$ в буфере с появлением новых значений *i*-го показателя и записью их в память не очищать буфер от старых значений, то могут возникнуть искажения в данных, аналогичные рассмотренным ранее.

Для решения этих проблем можно помещать все перезаписываемые значения в буфер, и тогда не будет ни искажений, ни больших пропусков в данных. Но при этом ухудшается производительность за счет двойной записи (в основную память и в буфер). Пожертвуем производительностью. Дело в том, что сбор данных происходит периодически и занимает больше времени, чем запись в память. Поэтому можно позволить в данной ситуации двойную запись: это не замедлит работу системы в целом.

Примечание: для других условий функционирования системы сбора данных или в случае использования устройства памяти с ограниченным числом циклов перезаписи этот вариант реализации не подойдет и может потребовать модификации.

Технические детали

При перезаписи данных возникает еще одна небольшая чисто техническая задача — обеспечение корректности повторного использования памяти. Хранимые кортежи имеют переменную длину, поэтому при перезаписи нового кортежа поверх старых данных могут возникнуть разные ситуации:

- новый кортеж занимает столько же памяти, сколько старый, или меньше. Тогда никаких проблем не возникает: он просто записывается в память вслед за предыдущим и изменяется значение адреса первого кортежа *R*1;
- новый кортеж имеет длину, превышающую длину старого кортежа (в общем случае — превышающего суммарную длину *k* первых кортежей, подлежащих перезаписи). Тогда перед записью нового кортежа в память необходимо обработать все подлежащие перезаписи кортежи на предмет извлечения из них значений, подлежащих записи в буфер.

Окончательно последовательность записи данных в память будет выглядеть следующим образом:

1. Инициализировать переменные:
   — N — число датчиков;
   — NMAX — адрес конца области хранения данных;
   — B[N] — буфер, B[i]:=null, i = 1, ..., N; — K[N] — кэш, K[i]:=null, i = 1, ..., N;
   — E[N] — вектор погрешностей измерения показателей, i = 1, ..., N.
2. Установить указатели P1 на первый и PN на последний кортежи на начало области хранения данных (P1 := 0, PN := 0).
3. Снять показания датчиков D[i] и сформировать текущий кортеж *C*:
   Если K[i] := null,
   то установить K[i] := D[i] и добавить D[i] в кортеж,
   иначе если | K[i] − E[i] | > D[i],
   то установить K[i] := D[i] и добавить D[i] в кортеж;
   к.е.
   к.е.
   Вычислить LEN — длину кортежа *C*.
4. Если PN > P1 и PN + LEN <= NMAX,
   то записать кортеж *C* в память по адресу PN,
   установить PN := PN + LEN;
   иначе
   Если PN + LEN > NMAX,
   то PN : = P1:
   к.е.
   Считать кортеж *R* по адресу P1.
   Вычислить LEN1 — длину кортежа *R* по адресу P1.
   Цикл пока PN + LEN > P1
   Цикл по i = 1 до N
   Если кортеж *R* содержит показания i-го датчика,
   то если B[i]<>null или

кортеж *C* не содержит показания i-го датчика,
то B[i] := R[i];
к.е.
к.е.
к.ц.
P1 := P1 + LEN1;
Считать кортеж *R* по адресу P1.
Вычислить LEN1 — длину кортежа *R* по адресу P1
к.ц.
Записать кортеж *C* в память по адресу PN.
Установить PN := PN + LEN.
к.е.
5. Вернуться к п. 3.

Заключение

В статье была рассмотрена задача организации хранения данных в памяти системы автономных устройств, оснащенных датчиками и собирающими информацию о состоянии окружающей среды. Был предложен метод организации памяти в виде кольцевого буфера и двух вспомогательных структур, который позволит обеспечить экономное хранение данных в узлах сети сбора информации в условиях жестких ограничений на ресурсы.

К сожалению, невозможно получить однозначную оценку времени и количества энергии, которые система будет тратить на запись данных в память. В зависимости от того, когда начинают возникать пропуски в данных — в начале цикла перезаписи или в конце, — система будет тратить разное время и разное количество энергии на размещение данных в памяти. Поэтому возможны либо средние оценки, либо оценки в диапазоне.

Список литературы

1. **Восков Л. С., Курпатов Р. О.** Энергоэффективный комбинированный метод локализации в беспроводных сенсорных сетях // Датчики и системы. 2011. № 4. С. 42—45.
2. **Карпов В. Э.** Коллективное поведение роботов. Желаемое и действительное // Современная мехатроника. Сб. науч. тр. Всероссийской научной школы (г. Орехово-Зуево, 22—23 сентября 2011). 2011. С. 35—51.
3. **Сергиевский М.** Беспроводные сенсорные сети // КомпьютерПресс, 2007. № 8. URL: http://www.compress.ru/article.aspx?id=17950
4. **Кузнецов С. Д., Гринев М. Н.** Ландшафт области управления данными: аналитический обзор. URL: http://citforum.ru/database/data_management_overview/5.shtml

---

УДК 004.272

Ю. Ф. Опадчий, д-р техн. наук, проф.,
Е. В. Чумакова, канд. физ.-мат. наук, доц.,
Российский государственный технологический университет имени К. Э. Циолковского "МАТИ",
e-mail: ekat.v.ch@rambler.ru

Методика разработки параллельных вычислительных систем обработки информации в режиме реального времени

На основе проведенного анализа тенденций развития бортовых вычислительных систем, работающих в реальном времени, предложено создание единой интегрированной вычислительной среды при функционально-ориентированном принципе построения комплекса бортового оборудования. Разработана методика построения вычислительной системы, которая позволяет проектировать систему параллельно работающих унифицированных вычислительных модулей, реализуемых на ПЛИС.

Ключевые слова: *интегрированная вычислительная среда, параллельная обработка информации, программируемые логические интегральные схемы (ПЛИС)*

Неотъемлемой частью любой современной системы реального времени является вычислительная система, круг решаемых задач которой постоянно расширяется. С увеличением области применения вычислительных систем (от управления до реализации экспертных систем с элементами искусственного интеллекта) возрастают требования к их техническим характеристикам. Современная вычислительная система (ВС) должна иметь высокое быстродействие, т. е. обеспечивать обработку больших массивов информации в режиме реального времени, при сохранении высокой живучести всей системы.

Проводимые исследования показали, что искомое решение должно формироваться не столько в области повышения технических характеристик, качества и эффективности функционирования отдельных элементов ВС, сколько в области поиска новых концепций и следующих возможных подходов к разработке архитектур:

- формирование семейства устройств — унифицированных модулей (*common modules*), с помощью которых может быть реализовано более 90 % программируемых и аппаратных функций;
- реализация отдельного общего модуля в виде одной или нескольких сверхбольших интегральных схем и построение вычислительных средств функциональных подсистем из той или иной

совокупности общих модулей (современный уровень развития электронных технологий это позволяет);
- построение программного обеспечения по-прежнему по модульному принципу из общих и специализированных программных модулей;
- организация технического обслуживания на основе сменных блоков LRU (*Line Replacable Unit*) и при этом использование в качестве единицы "физической" архитектуры общего унифицированного модуля.

ВС в системах реального времени должны базироваться на функционально-ориентированной структуре, организованной по типу интегрированной вычислительной среды (ИВС) [1]. При этом должно отсутствовать регулярное распределение средств вычислительной техники по функциональным подсистемам и информационным каналам, что предопределяет значительное повышение живучести всей системы. Концепция аппаратной интеграции требует построения ИВС, которая обеспечит независимость программ от используемых аппаратных средств и расширит функциональные возможности системы на основе комплексной обработки информации.

Цель исследования — разработка принципов организации вычислительного процесса и методики проектирования алгоритмов функционирования вычислительных модулей в рамках единого вычислительного пространства.

Разработка принципов организации вычислительного процесса

Вполне очевидно, что решение такой задачи возможно только при полном пересмотре самих принципов построения аппаратных средств. Существуют два традиционных подхода к решению поставленной задачи. Первый предполагает реализацию аппаратной части с использованием универсальных микропроцессоров и организацию требуемых алгоритмов обработки информации программными средствами. Этот подход хорошо зарекомендовал себя при разработке ЭВМ общего назначения и позволяет реализовать ИВС, в большей части отвечающую сформулированным выше требованиям. Однако основным недостатком такого решения, несмотря на достигнутые в настоящее время тактовые частоты работы микропроцессоров на уровне нескольких гигагерц, является недостаточное быстродействие при необходимости обработки больших массивов информации в режиме реального времени. Значительно большее быстродействие имеют специализированные системы, сама архитектура которых ориентирована на выполнение заданного алгоритма обработки информации (системы на жесткой логике). Этот принцип широко применяют при построении так называемых сигнальных процессоров, используемых, например, в системах обработки изображения. Однако реализация такого подхода не удовлетворяет требованиям к ИВС нового поколения, в частности, к открытости и адаптивности архитектуры к различным вариантам применения и аппаратной интеграции.

Решение задачи построения ИВС требует совмещения достоинств обоих известных подходов, т. е. возможности гибкого изменения алгоритмов работы системы при адаптации ее структуры к выполнению вычислений на аппаратном уровне. Основой для реализации такого подхода может служить использование новой элементной базы, так называемых программируемых логических интегральных схем (ПЛИС). Структура ПЛИС может гибко адаптироваться к заданному алгоритму обработки информации, причем сама адаптация может выполняться непосредственно во время работы устройства. Такая реализация системы на кристалле получила название SoPC (System on a programmable chip). Компоненты SoPC, алгоритм работы которых задается на одном из языков описания аппаратуры (HDL — Hardware Description Language), разрабатываются отдельно и хранятся в виде файлов параметризируемых модулей. Окончательная структура SoPC-микросхемы выполняется на базе этих "виртуальных компонентов" с помощью программ систем автоматизации проектирования (САПР) электронных устройств.

Благодаря стандартизации и применению HDL можно объединять "виртуальные компоненты" от разных разработчиков. Существующие ПЛИС позволяют создавать на кристалле некоторую однородную программируемую среду, с возможностью ее конфигурирования для решения конкретных задач. Это обеспечивает, с одной стороны, максимальное быстродействие — возможность одновременного решения большого числа задач в режиме реального времени, и, с другой стороны, максимальную живучесть — способность выполнять функции в полном объеме при частичном повреждении системы. При этом вся система может быть выполнена на основе однотипных унифицированных модулей, так как на основе ПЛИС могут быть разработаны сменные блоки LRU. Смена алгоритма работы такого блока проводится их соответствующей реконфигурацией либо при включении, либо непосредственно в процессе работы. Система, реализованная на ПЛИС, не привязана к конкретной аппаратной архитектуре, а ее аппаратная гибкость позволяет выполнять модульное наращивание системы в рамках одной унифицированной архитектуры. Такие системы приближаются к оптимальным по размерам, потребляемой мощности и внутрисистемным задержкам распространения сигналов.

Повышать производительность ВС можно не только совершенствованием аппаратных средств, но и использованием более эффективных алгоритмов работы, позволяющих без увеличения тактовой

частоты работы аппаратных средств значительно поднять эффективную производительность бортовых ВС. В настоящее время основными направлениями совершенствования алгоритмического обеспечения являются модульность и параллельность, т. е. алгоритмы обработки информации должны иметь модульную организацию, а также давать возможность распараллеливания процесса обработки данных.

Учитывая, что разрабатываемые системы имеют сложную структуру, содержащую большое число узлов, целесообразно ввести условие, согласно которому вычисления в любом узле этого вычислительного пространства проводятся с одинаковой (предельной) точностью и быстродействием, т. е. реализовать единое вычислительное пространство. Использование концепции единого вычислительного пространства, базирующегося на применении унифицированных модульных алгоритмов на основе ПЛИС, позволит в значительной степени повысить быстродействие обработки информации в системах реального времени.

Методика проектирования бортовых вычислительных модулей параллельной обработки информации

На основе исследования алгоритмов цифровой обработки системы навигации летательного аппарата сформулирована последовательность этапов проектирования единого вычислительного пространства, базирующегося на использовании ПЛИС. Проектирование функциональных узлов цифровой обработки информации бортовой системы, основанных на использовании ПЛИС, выполняют в следующей общей последовательности:

- постановка задачи — определение исходных данных, таких как требования к техническим характеристикам ВС (быстродействие, точность и т. п.), доступные аппаратные ресурсы ПЛИС;
- выбор алгоритмов — определение последовательности алгоритмов, удовлетворяющих требованиям концепции единого вычислительного пространства и оптимальности с точки зрения их реализации на ПЛИС. К таким алгоритмам можно отнести те, которые имеют параллельную или последовательно-параллельную структуру, а также модульную организацию. Модуль должен состоять из определенной последовательности математических операций и функций, а все вычисления сводятся к вычислению этого модуля нужное число раз.
- реализация вычислений математических операций и функций в бортовой системе, включающая [2, 3]:
  - — выбор формата и разрядности представления используемых чисел;
  - — выбор алгоритмов работы и программ для ПЛИС, обеспечивающих аппаратную реализацию вычисления основных математических операций;
  - — выбор алгоритмов работы и программ для ПЛИС, обеспечивающих аппаратную реализацию вычисления требуемых математических функций (критерием выбора служат время вычисления и затрачиваемые аппаратные ресурсы ПЛИС, которые можно оценить с применением САПР);
- определение принципов организации вычислительного процесса, разработка алгоритма его работы и программ инициализации ПЛИС [4, 5];
- выбор алгоритма управления модулем обработки информации ВС (алгоритм одноступенчатой или двухступенчатой выборки данных), разработка реализующих его программ инициализации ПЛИС [4, 5];
- моделирование с использованием САПР разработанных алгоритмов и программ и определение их соответствия исходным данным на разработку.

Обсуждение результатов

Результатом применения методики является получение удовлетворяющих исходным требованиям структур вычислительного модуля и комплекса программ, предназначенных для формирования на ПЛИС единого вычислительного пространства. Полученный с помощью предложенной методики вычислительный модуль состоит из семи параллельно работающих унифицированных модулей (см. рисунок), число которых может меняться в зависимости от алгоритмов цифровой обработки ВС.

Для организации вычислений в угломерном канале необходимо построить 23 блока универсальных вычислительных структур различного типа, для которых максимально возможное число необходимых входных сигналов равно 16. Управление работой этих модулей осуществляется с помощью управляющего автомата, реализующего двухступенчатую выборку данных. Использование двухступенчатой выборки данных позволит наращивать алгоритмическое обеспечение системы без изменения структуры самих унифицированных вычислительных модулей, а также потребует меньших аппаратных затрат по сравнению с одноступенчатой выборкой данных (в 1,5 и более раз) [4]. Производительность всего модуля больше требуемой в 32 раза, а при сравнении с многопроцессорной системой (с тактовой частотой 40 МГц) последняя уступает в производительности в 4 раза. В качестве базовых для оценки использовали ПЛИС, имеющие максимальные задержки (APEX20KE фирмы *Altera*). Полученное увеличение производительности в значительной степени зависит от выбранных алгоритмов обработки информации, однако предложенная методика может быть использована при проектировании любых ВС реального времени.

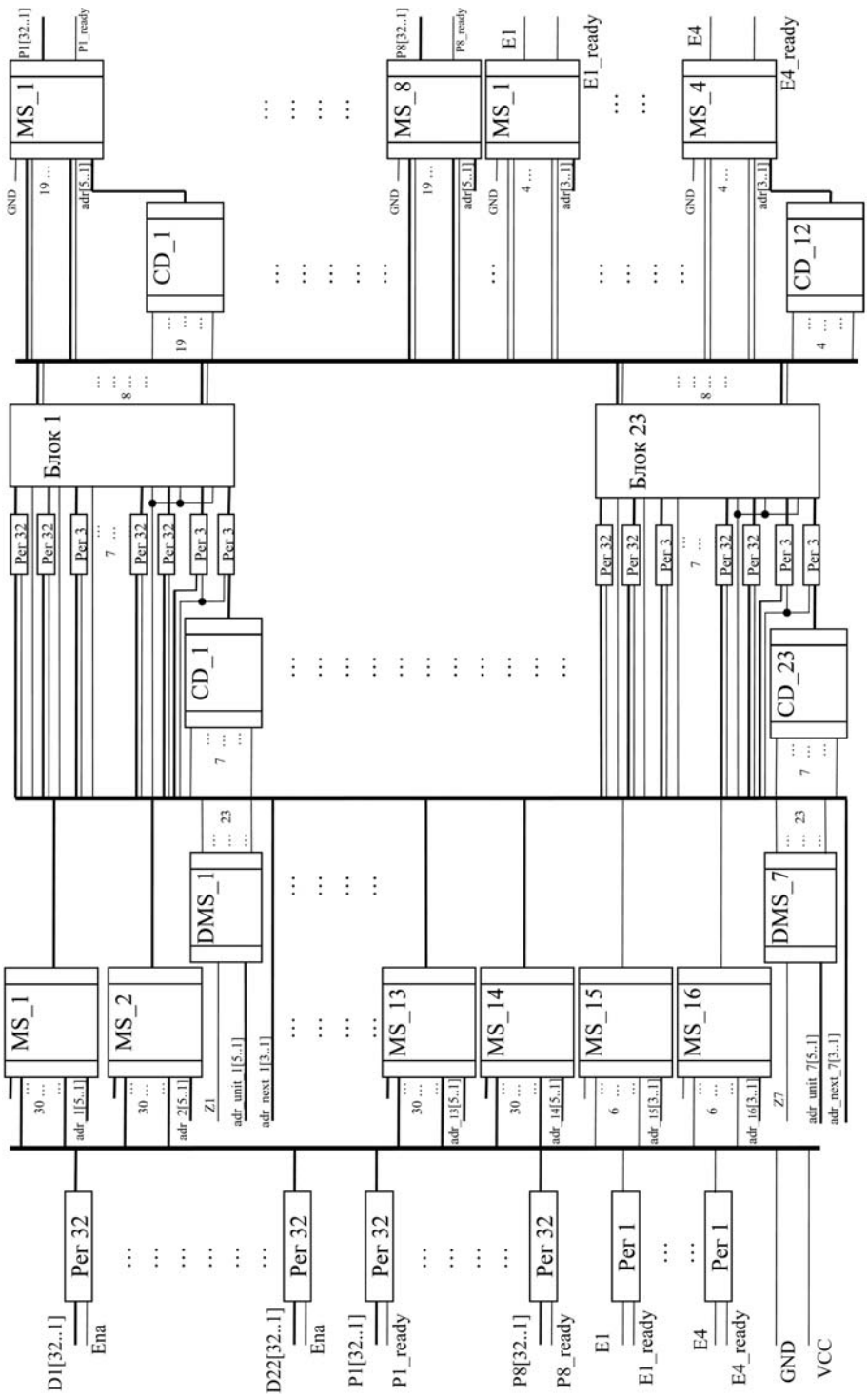


Схема вычислительного модуля с двухступенчатой выборкой

Выводы

Результаты исследований вычислительного процесса обработки информации в системах реального времени, а также алгоритмов вычисления элементарных математических операций и функций позволяют разработать общую методику проектирования устройств данного класса. Методика ориентирована на применение ПЛИС, что позволяет перейти к реализации принципиально новой, функционально-ориентированной структуры ВС, организованной по типу интегрированной вычислительной среды, и предопределяет значительное расширение функциональных возможностей и повышение живучести всей системы. Технологии, описанные в методике, могут быть использованы при проектировании любых вычислительных систем реального времени.

Список литературы

1. **Павлов А. М.** Принципы организации бортовых вычислительных систем перспективных летательных аппаратов // Мир компьютерной автоматизации. 2001. № 4. С. 27—36.

2. **Чумакова Е. В.** Алгоритмы операций умножения и деления для реализации на ПЛИС // Проектирование и технология электронных средств. 2005. № 2. С. 54—57.

3. **Опадчий Ю. Ф., Чумакова Е. В.** Реализация на ПЛИС вычисления элементарных математических функций // Проектирование и технология электронных средств. 2005. № 4. С. 7—12.

4. **Чумакова Е. В.** Алгоритмы работы функциональных узлов цифровой обработки информации, основанных на ПЛИС // Электронный журнал "Исследовано в России". 2009. 070. С. 923—930. URL: http://zhurnal.ape.relarn.ru/articles/2009/070.pdf (дата обращения: 10.04.2010).

5. **Кораблин Ю. П., Куликова Н. Л., Чумакова Е. В.** Алгоритмы обработки цифровой информации для реализации на ПЛИС. М.: ИНЭК, 2009. 40 с.

УДК 004.31

И. П. Норенков, д-р техн. наук, проф.,
МГТУ им. Н. Э. Баумана

Алгоритм упорядочения модулей в синтезируемых учебных пособиях

Представлены модель базы образовательных ресурсов и алгоритм упорядочения обучающих фрагментов (модулей) при формировании электронных учебных пособий в соответствии с технологией разделяемых единиц контента.

Ключевые слова: *электронное учебное пособие, маршрут обучения, упорядочение модулей*

Введение

В настоящее время для автоматизированного синтеза электронных учебных пособий применяют технологии SCORM [1] или ТРЕК (технологии разделяемых единиц контента) [2], в соответствии с которыми электронные учебные пособия (ЭУП) компилируются из предварительно разработанных фрагментов, называемых модулями (или разделяемыми единицами контента) и представляющих содержательную часть базы учебных материалов (БУМ). Особенностью технологии ТРЕК является использование для автоматизации синтеза пособий понятийного каркаса БУМ в виде онтологий рассматриваемых предметных областей. Целями применения этих технологий являются сокращение материальных и временных затрат на разработку ЭУП и способствование индивидуализации обучения.

Индивидуализация обучения обеспечивается адаптацией синтезируемых ЭУП к потребностям и уровню предварительной подготовки обучаемого, заключающейся в индивидуальном подборе как множества изучаемых понятий, так и модулей, разъясняющих эти понятия. Однако для эффективной адаптации пособий необходимо в БУМ иметь для каждого понятия несколько разъясняющих модулей, различающихся уровнями подробности, сложности, глубины и тому подобными характеристиками изложения. Это требование ведет к росту размеров БУМ и многовариантности решения задач как выбора подмножества модулей для конкретного ЭУП, так и упорядочения отобранных модулей в составе ЭУП.

Метод оптимального решения первой из названных задач предложен в работе [3]. Он основан на поиске оптимального подграфа в И/ИЛИ-графе, моделирующем БУМ. Алгоритм решения второй из указанных задач предлагается в данной статье.

Модели БУМ и ЭУП

Моделью БУМ в задаче оптимизации индивидуальной траектории обучения и синтеза ЭУП, поддерживающего эту траекторию, является И/ИЛИ-граф, в котором И-вершины соответствуют модулям, а ИЛИ-вершины — концептам (понятиям). Дуга направлена от И-вершины модуля A к ИЛИ-вершине концепта B, если концепт B определен и пояснен в модуле A. Дуга направлена от ИЛИ-вершины концепта C к И-вершине модуля A, если концепт C использован в модуле A и необходим для понимания содержимого этого модуля (рис. 1).

Для синтеза ЭУП нужно задать множество целевых концептов, которые должны быть изучены с помощью синтезируемого пособия. Известны также исходные концепты, т. е. концепты, уже изученные обучаемым. Тогда моделью ЭУП является подграф И/ИЛИ-графа БУМ, содержащий пути от вершин исходных понятий к вершинам целевых понятий и оптимальный по заданному критерию. Если отмечать вхождение j-й И-вершины в подграф ЭУП как $k_j = 1$, q-й ИЛИ-вершины как $m_q = 1$, а их невхождение в ЭУП как $k_j = 0$ и $m_q = 0$, то модель ЭУП представляется в виде логического уравнения

$$\&k_j = 1, \quad j \in \mathbf{J}, \tag{1}$$

где $\mathbf{J}$ — множество номеров целевых концептов, и для каждой p-й ИЛИ-вершины справедливо выражение

$$k_p = Vm_i \& k_t, \quad i \in \mathbf{I}_p,\ t \in \mathbf{T}_i, \tag{2}$$

$\mathbf{I}_p$ — множество номеров модулей, в которых определяется p-й концепт; $\mathbf{T}_i$ — множество номеров концептов, используемых в i-м модуле. Последовательная подстановка (2) в уравнение (1) с учетом того, что переменные исходных концептов $k_t = 1$, приводит к получению дизъюнктивной нормальной формы, в которой каждая элементарная конъюнкция представляет одно из возможных решений задачи.

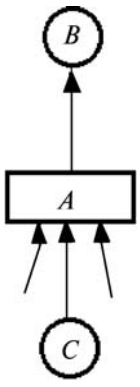


Рис. 1

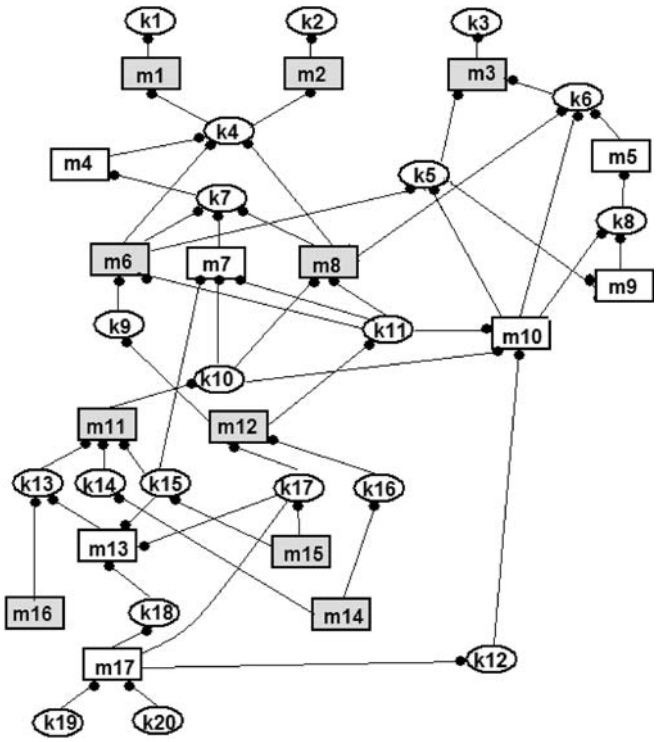


Рис. 2

Постановку задачи и ее решение иллюстрирует пример И/ИЛИ-графа из работы [3] (рис. 2, на котором ИЛИ-вершины показаны в виде овалов, И-вершины — в виде прямоугольников; исходные концепты с инцидентными им дугами на рисунке не показаны).

Пусть пользователь (это может быть как преподаватель, так и сам обучаемый) указывает в качестве цели изучение концептов k1, k2 и k3. На основе модели (1)—(2) найдено множество альтернативных решений, из которых по критерию минимального числа модулей в ЭУП выбрано подмножество

$$\mathbf{M}_{\text{орт}} = \{\text{m1, m2, m3, m6, m8, m11, m12, m14, m15, m16}\}, \quad (3)$$

модули которого на рис. 2 показаны затененными.

Алгоритм упорядочения модулей

Синтез ЭУП завершается упорядочением модулей в составе ЭУП. Алгоритм упорядочения модулей может быть основан на стратегии поиска "в ширину" или "в глубину".

Стратегия "в ширину" подразумевает включение в маршрут модулей по принципу: чем раньше у модуля определены входы (входные концепты), тем раньше этот модуль входит в маршрут (упорядоченную последовательность модулей в составе ЭУП). Этот принцип реализуется на основе ранжирования модулей, при котором модули, не имеющие неизученных входов, и выходы этих модулей получают ранг 0. Если старший из рангов входов модуля X имеет ранг r, то модуль X и его выходы относят к рангу $r + 1$. Принцип "в ширину" подразумевает, что модули включаются в маршрут обучения в порядке увеличения их рангов. В нашем примере один из возможных маршрутов, полученных в соответствии со стратегией "в ширину", представляет собой следующую последовательность модулей:

{m16, m15, m14, m12, m11, m8, m6, m3, m2, m1},

которая собственно и была представлена в подмножестве (3).

В большинстве случаев предпочтение отдается стратегии "в глубину", так как при этом наблюдается более логичное изложение, лучшая концентрация в учебнике родственного по семантике учебного материала.

Алгоритм преобразования полученного оптимального множества $\mathbf{M}_{\text{орт}}$ в маршрут обучения по принципу "в глубину" может быть представлен в следующем виде.

Из числа модулей, входящих в $\mathbf{M}_{\text{орт}}$, образуется последовательность, состоящая из двух подпоследовательностей **M**1 и **M**2, разделенных маркером $ и состоящих из переменного числа позиций. **M**1 включает модули, уже упорядоченные на данный момент исполнения алгоритма, **M**2 — модули, уже вовлеченные в обработку, но еще не упорядоченные.

В начале алгоритма **M**1 — пустое множество, **M**2 включает модули, не имеющие иных выходов, кроме целевых.

Для нашего примера исходное состояние **M**1 = = пусто и **M**2 = {m1, m2, m3}.

Алгоритм включает следующие операции:

1. Если **M**2 = пусто, то **M**1 — искомый маршрут, и конец алгоритма.

2. Для первого стоящего после $ модуля m_i записываются модули, находящиеся с m_i в отношении "предыдущий/последующий".

3. Если первый после $ модуль определен (все входные концепты относятся к исходным или к уже определенным), то он переписывается в **M**1 непосредственно перед $.

4. Переход к шагу 1.

Результат исполнения этого алгоритма зависит от того, в каком порядке указываются целевые модули в исходном состоянии подмножества **M**2 и записываются модули на шаге 2 алгоритма. Один из возможных вариантов маршрута, сформированного на основе стратегии "в глубину" для рассматриваемого примера, есть последовательность:

{m14, m15, m12, m6, m1, m2, m16, m11, m8, m3}.(4)

Пооперационно исполнение алгоритма формирования последовательности (4) характеризуется следующими результатами:

$ m6 m1 m2 m3;
$ m12 m6 m1 m2 m3;
$ m14 m12 m6 m1 m2 m3;
m14 $ m15 m12 m6 m1 m2 m3;
m14 m15 m12 m6 m1 m2 $ m3;
m14 m15 m12 m6 m1 m2 $ m8 m3 (уже включенные в **M**1 модули повторно в **M**2 не включаются, поэтому здесь после $ модуль m6 отсутствует);
m14 m15 m12 m6 m1 m2 $ m11 m8 m3;
m14 m15 m12 m6 m1 m2 $ m16 m11 m8 m3;
m14 m15 m12 m6 m1 m2 m16 m11 m8 m3 $.

Достоинством представленного алгоритма является возможность его обобщения на случай наличия контуров в сети модулей. В оператор 2 обобщенного алгоритма нужно добавить правило: если претендент на вписывание между $ и модулем m_j уже имеется в **M**2, то вписывание не выполняется, что соответствует разрыву и игнорированию связи между претендентом и модулем m_j.

В качестве искомой последовательности модулей может быть принят любой вариант, сформированный по представленному алгоритму. Для выбора оптимального маршрута среди маршрутов, сформированных по принципу "в глубину", целесообразно использовать генетические алгоритмы. Каждый вариант маршрута задается хромосомой, в которой гены соответствуют вершинам И/ИЛИ-графа, а аллелями являются значения приоритетов на вписывание в **M**2 в виде чисел в интервале $[1{:}N]$, где $N \gg l$, l — число вершин графа. В качестве максимизируемой целевой функции целесообразно принять суммарную связность модулей:

$$S = \sum_{i \in \mathbf{E}} ((1/(s_i + 1))/n,$$

где связность s_i пары модулей A и B есть число модулей, находящихся между A и B в маршруте обучения, если A — модуль, в котором определяется некоторый концепт, B — модуль, в котором этот концепт используется, n — число семантически связанных пар модулей, **E** — множество таких пар. Например, для маршрута (3) $S = 0{,}52$, а для маршрута (4) $S = 0{,}73$ при $n = 10$. При росте размера задачи связность S при поиске "в ширину" стремится к значению r_{max}/n, где r_{max} — максимальный ранг среди вершин графа, и преимущества поиска "в глубину" становятся более заметными.

Заключение

Современные технологии синтеза электронных учебных пособий позволяют автоматически отбирать из имеющихся баз учебных материалов подмножества отдельных обучающих фрагментов, отвечающих образовательным потребностям пользователя. Собственно учебное пособие создается упорядочением этого подмножества. Алгоритм упорядочения отобранных модулей в составе пособия представлен в данной статье.

Список литературы

1. **SCORM.** Shareable Content Object Reference Model. 2d Edition. — Advanced Distributed Learning, 2004.
2. **Норенков И. П.** Технологии разделяемых единиц контента для создания и сопровождения информационно-образовательных сред // Информационные технологии. 2003. № 8. С. 34—39.
3. **Норенков И. П., Соколов Н. К.** Синтез индивидуальных маршрутов обучения в онтологических обучающих системах // Информационные технологии. 2009. № 3. С. 74—77.

Информация

11th IEEE EAST-WEST DESIGN & TEST SYMPOSIUM (EWDTS 2013)

состоится в городе Ростов-на-Дону, Россия, 27—30 сентября 2013 года

Цель симпозиума — расширение международного сотрудничества и обмен опытом между ведущими учеными Западной и Восточной Европы, Северной Америки и других стран в области автоматизации проектирования, тестирования и верификации электронных компонентов и систем. Симпозиум проводится, как правило, в странах бассейнов Черного и Балтийского морей, Центральной Азии. Оргкомитет приглашает ученых, аспирантов и студентов принять участие в работе EWDTS'13.

Симпозиум будет проходить в Ростове-на-Дону — крупнейшем научном и образовательном центре Южного федерального округа России.

Важные даты: **Срок подачи докладов:** 15 июля 2013 г.
Итоги рецензирования: 1 августа 2013 г.

Детальная информация и регистрация докладов: http://www.ewdtest.com/conf

Адрес оргкомитета: Проф. Владимир Хаханов, кафедра Автоматизации проектирования вычислительной техники Харьковского национального университета радиоэлектроники, пр. Ленина 14, Харьков, 61166, Украина. Тел.: +380-57-702-13-26, E-mail: hahanov@kture.kharkov.ua, www.ewdtest.com/conf/

УДК 621.396

В. В. Пехтерев, магистр,
e-mail: PekhterevVV@mpei.ru,
С. В. Вишняков, канд. техн. наук, доц.,
e-mail: vsv@emc.mpei.ac.ru,
М. К. Чобану, д-р техн. наук, проф.,
e-mail: tchobanou@yahoo.com,
ФГБОУ ВПО Национальный
исследовательский университет "МЭИ"

Адаптивная триангуляция и сжатие изображений

Предложен алгоритм создания адаптивной сетки на основе треугольных элементов, предназначенный для применения в эффективном алгоритме распознавания объектов, обнаружения движения и сжатия видеосигналов, который позволяет достигать сравнительно высокого качества кодированного изображения (при сжатии с потерями). Исследованы характеристики предлагаемого алгоритма в части сжатия изображения, управления густотой сетки.

Ключевые слова: *триангуляция, адаптивная сетка, сжатие изображений и видео*

Введение

Эффективное кодирование видеосигналов высокого и сверхвысокого разрешения в настоящее время является одним из наиболее актуальных направлений развития методов цифровой обработки сигналов [1, 2]. Целью кодирования является, прежде всего, сжатие с потерями видеосигналов для передачи через канал с ограниченной полосой. Основными подходами к сжатию видео, реализованными в настоящий момент, являются:

- выделение базовых кадров (*I-frame* [3]) с кодированием разностного сигнала последующих кадров и базового (есть иные варианты);
- разделение кадров на блоки меньшей размерности в целях параллельной обработки для сжатия и последующей интерполяции при восстановлении видеосигнала [4].

Предлагаемый ниже алгоритм находится, в целом, в русле этих идей. При этом предполагается повышение качества кодирования и эффективности предсказания последующих кадров. Недостатком существующего подхода является, в частности, разделение кадров на прямоугольные блоки [5]. Отметим, что границы этих блоков зачастую не соответствуют естественным границам изображаемых объектов. Для преодоления такого недостатка предлагалось применять равномерные и неравномерные гексагональные [6] и треугольные сетки [7, 8]. Заметим, что равномерные сетки позволяют использовать регулярные преобразования и регулярную нумерацию элементов разбиения (в этом случае нет необходимости хранить какую-либо информацию об элементах сетки, например координаты вершин), что увеличивает потенциальную скорость компрессии и декомпрессии кадров. Однако повышение качества кусочно-непрерывной аппроксимации изображения может быть достигнуто лишь с применением адаптивных сеток, прежде всего, на основе треугольных элементов.

Повышение эффективности кодирования видеосигналов может быть достигнуто за счет учета внутрикадровых перемещений (масштабирования или поворота) отдельных объектов или частей объектов, что, с одной стороны, требует применения алгоритмов распознавания образов, но, с другой стороны, позволяет резко увеличить коэффициент сжатия последующих кадров [2]. Другим перспективным направлением является выделение объектов, имеющих определенный характер заполнения — тоновую заливку, текстуру, шумоподобный сигнал — для выбора наилучшего метода предварительной обработки части изображения и его сжатия [5].

Представляется целесообразным использование треугольной сетки для описания исходного изображения (кадра) и при решении указанных задач. С одной стороны, это предполагает *адаптацию* сетки к особенностям изображения (что существенно для выделения объектов и обнаружения движения). С другой стороны, это позволяет использовать технологии деформации сетки (*mesh morphing*), хорошо зарекомендовавшие себя при решении задач упругой деформации и разрушения методом конечных элементов, для эффективного описания перехода между кадрами и для компенсации движения [9].

В настоящее время разрабатывается ряд методов обработки изображений и видеосигналов на основе триангуляции [8]. Необходимо выделить следующие этапы в решении задачи:

- предобработка изображения (кадра) — удаление шумов, сглаживание, применение адаптивных фильтров;
- вычисление критерия, определяющего необходимость более или менее густого разбиения, по-

ложение узлов (вершин) и треугольников (элементов разбиения);
- построение собственно разбиения;
- сжатие изображения.

Необходимо подчеркнуть важность первого этапа, поскольку построение адаптивной сетки во всех случаях является нелинейной операцией (всегда присутствует выбор в соответствии со значением некоторого критерия), и шумы, а также незначительные искажения изображения способны вызвать излишнее сгущение сетки и, соответственно, снизить эффективность алгоритма кодирования.

Второй этап часто реализуют с применением рекурсивных алгоритмов, когда сетка сгущается или разрежается в соответствии со значением некоторого критерия [10]; на каждом шаге обрабатывается лишь часть изображения. Необходимо, однако, отметить, что выполнение рекурсивного сгущения/разрежения треугольной сетки, как правило, требует значительных вычислительных затрат. Вместе с тем, требование связности сетки приводит к необходимости учитывать разбиение отдельного элемента при перестроении части сетки, что усложняет распараллеливание вычислений.

Третий этап, как правило, основан на выполнении триангуляции Делоне (могут варьироваться некоторые детали — порядок обхода, учет качества сетки и т. п.). Рациональным видится использование специализированных алгоритмов, учитывающих дискретный характер решаемой задачи.

Сжатие содержимого элемента (группы элементов) может осуществляться различными способами, которые условно можно разделить на две группы:
- работа с одномерной последовательностью отсчетов (пиксели, содержащиеся в элементе, размещаются в одномерном массиве);
- применение различных преобразований изображений, например, двумерного дискретного косинусного или вейвлет-преобразования [11].

Следует отметить, что применение многомерных (двумерных) методов выглядит предпочтительнее за счет использования большего числа корреляционных связей, что, очевидно, должно быть эффективнее в смысле устранения избыточности в рассматриваемом сигнале.

Комплексный анализ проблемы позволяет предположить, что эффективный алгоритм сжатия на основе адаптивной триангуляции должен удовлетворять следующим требованиям:
- предобработка должна учитывать возможности сохранения шумоподобных составляющих на этапе сжатия, т. е. желательной является очистка изображения от *высокочастотных составляющих*, но при сохранении существенных границ объектов;
- формирование сетки следует проводить за *один проход*, без рекурсии;
- необходимо строить сетку сравнительно *высокого качества* (не допуская появления элементов с острыми углами);
- управление соотношением качества и степени сжатия следует осуществлять в первую очередь, *ограничивая размер* элементов;
- алгоритм построения сетки целесообразно сформировать таким образом, чтобы сетка *однозначно строилась* на основе заданной совокупности узлов.

Рациональное сочетание изложенных принципов позволяет создать метод кодирования, обладающий рядом преимуществ: высокой адаптивностью, управляемым коэффициентом сжатия при заданном качестве, низкой вычислительной сложностью, особенно при распараллеливании процесса обработки сигналов.

Этап предобработки изображения

Целесообразной является предварительная очистка изображения от шума, если такая обработка допускается по условиям дальнейшей работы со сжатым изображением. Для подавления шумов можно использовать метод [12], применяемый для обработки изображений [13], в том числе для данных аэрофотосъемки и съемки из космоса [14]. Применение метода позволяет достичь значительного улучшения соотношения сигнал-шум при сохранении четких границ объектов.

Для отработки этапов распознавания контуров и создания сетки, а также сжатия статических изображений, использовались следующие изображения:

1) телевизионный кадр высокой четкости — городской пейзаж, 1920 × 1080, 24 бит/пиксель, гистограмма яркости имеет максимум при значении, равном 64, а также всплеск в диапазоне 250...255, имеются значительные однородные участки;

2) изображение "Лена", портрет, 512 × 512, 24 бит/пиксель, широко применяемое тестовое изображение, гистограмма по каналу яркости имеет многоэкстремальный характер в диапазоне 30...220, имеются значительные однородные участки, а также текстурированные области, выражен аддитивный шум;

3) изображение "Озеро", пейзаж, 512 × 512, 24 бит/пиксель, известное тестовое изображение, гистограмма по каналу яркости имеет выраженные экстремумы в 55, 180 и 200, имеются значительные однородные участки, а также текстурированные области, выражен аддитивный шум;

4) изображение "Ручей", пейзаж, 3264 × 2448, 24 бит/пиксель, цифровая фотография, гистограмма по каналу яркости имеет почти равномерный характер в диапазоне 30...200, не содержит сколько-нибудь значительных однородных областей.

Результаты обработки изображений без/с подавлением шумов приведены в табл. 1. Для зашумленного тестового изображения 2 при заданной

Таблица 1

Влияние предварительного подавления аддитивного шума на распознавание контуров и создание сетки

Изображение	Обработка	Число контуров/ в % к исходному	Число треугольников/ в % к исходному
1	Исходное $\sigma^2 = 3$ $\sigma^2 = 10$	71406 70607/98,8 69559/97,3	24975 24497/98,1 23896/95,7
2	Исходное $\sigma^2 = 3$ $\sigma^2 = 10$	363 331/91,2 307/84,53	3188 2975/93,3 2852/89,5
3	Исходное $\sigma^2 = 3$ $\sigma^2 = 10$	12160 11779/96,8 11205/92,1	5717 5506/96,3 5088/89,0
4	Исходное $\sigma^2 = 3$ $\sigma^2 = 10$	474200 472358/99,6 468254/98,7	23821 23578/99,9 23344/98,0

дисперсии шума $\sigma^2 = 3$ получено на 10 % меньше контуров и на 7 % меньше треугольников сетки, чем для необработанного изображения, а при дисперсии шума $\sigma^2 = 10$ — на 15 % и 10 % меньше соответственно; при этом субъективная оценка качества выделения контуров и построения сетки не меняется.

Приведенные результаты свидетельствуют о том, что предлагаемый ниже подход к построению адаптивной сетки обладает высокой устойчивостью к аддитивному шуму в исходном изображении.

Выделение характерных особенностей изображения — контуров, является следующим этапом построения сетки. Поскольку выделение контуров и объектов основано на работе со скалярными полями, используется одно из следующих представлений (в модели *YUV*): в оттенках серого используется известное представление $Y = 0{,}299\,Red + 0{,}587\,Green + 0{,}114\,Blue$ через компоненты красного (*Red*), зеленого (*Green*) и синего (*Blue*); по каждой компоненте в отдельности все особенности, обнаруженные хотя бы по одному каналу, используются для построения общей сетки:

- вычисляется разностный сигнал (градиент) с простейшим крестообразным шаблоном $D_x = \begin{bmatrix} 1 \\ -1 \end{bmatrix} Y$; $D_y = Y[1\ -1]$;
- проводятся сглаживание разностных сигналов, дополнительное подавление шумовых составляющих. Для этого предлагается использовать центрально-симметричный фильтр с конечной импульсной характеристикой (КИХ) и Гауссовой характеристикой (на основании экспериментальных данных выбрано среднеквадратичное отклонение, равное 2) с носителем 5 × 5;
- на основе градиента поточечно вычисляются так называемые векторы ориентации — комплексные числа, модуль указывает на выраженность ориентации, а аргумент — направление. Для расчета применяется методика, изложенная в работе [15]:

$$O(x, y) = J_{xx}(x, y) - J_{yy}(x, y) + jJ_{xy}(x, y),$$

где $J_{\vartheta\xi}(x, y) = \iint w(p - x, q - y) D_\vartheta D_\xi dpdq$; w — весовая (Гауссова) функция;

- Гауссов фильтр выступает также в роли детектора контуров; для оценки направлений контуров используются восемь дискретных направлений, соответствующих соседним пикселям изображения;
- к разностным сигналам применяется адаптивная фильтрация с использованием предварительно синтезированного банка из восьми КИХ-фильтров с носителем 5 × 5. Производящим является фильтр с Гауссовой характеристикой в "поперечном" сечении и с постоянной характеристикой в "продольном"; банк фильтров формируется поворотом импульсной характеристики (ИХ) производящего фильтра на углы (45°, 90°, ..., 315°). Фильтр выбирается из банка в соответствии с аргументом вектора локальной ориентации, рассчитанного на предыдущем этапе;
- адаптивная фильтрация применяется к векторному полю локальной ориентации — используется описанный выше банк фильтров и банк фильтров с обращенной ИХ. Также при фильтрации осуществляется нормировка результирующих сигналов (для каждого банка фильтров в отдельности) в пределах носителя фильтра. Вычисляется разность полученных сигналов. Для подавления всплесков в области с низким модулем градиента выполняется поточечное перемножение полученного сигнала с сигналом $|D_x| + |D_y|$;
- выполняется трассировка контуров — после вычисления поточечного произведения на предыдущем этапе в результирующем сигнале (он и является критерием выделения контуров) выделяются области с положительным значением, для которых строится множество контуров. Каждая точка с положительным значением критерия рассматривается как возможная начальная точка контура, после чего проводится поиск точек с положительным критерием по соседству. Затем осуществляется кусочно-линейная аппроксимация полученных контуров, поскольку линейные участки контуров используются как ребра элементов сетки. На этом этапе задается пороговый коэффициент, ограничивающий длину участков линейной аппроксимации в зависимости от среднего расстояния точек контура до аппроксимирующей прямой.

Построение сетки, состоящей из треугольных элементов

На первом этапе в качестве узлов сетки используются точки начала и конца участков кусочно-линейной аппроксимации контуров. Проводится анализ взаимного расположения таких точек и в случае, если отдельные точки отстоят от ближайших на расстояние, превышающее некоторый порог h (величина h может интерпретироваться как максимальный шаг сетки), вводятся дополнительные узлы, не лежащие на контурах. Триангуляция осуществляется в соответствии с алгоритмом Делоне с ограничениями [16].

Алгоритм может быть дополнен анализом некоторых показателей качества сетки по аналогии с процедурами, применяемыми в конечно-элементном анализе. В частности, применение линейных функций для аппроксимации цветовых каналов внутри элементов сетки означает высокую чувствительность точности аппроксимации к форме треугольных элементов. Как правило, наличие острых (тупых) углов (менее 10...15° и более 150° соответственно) приводит к неудовлетворительному качеству аппроксимации даже для гладких функций [17). Как будет показано далее, низкое качество отдельных элементов сетки приводит к сложностям при перенумерации пикселей и выполнении дискретных преобразований изображения внутри элемента.

Важной особенностью предлагаемого подхода является однозначность создания сетки — для передачи изображения необходимо сохранять информацию лишь об узлах сетки, треугольники будут созданы и пронумерованы идентично сетке, созданной при кодировании.

Сжатие изображений

Сжатие изображения достигается комбинацией трех составляющих:

- кусочно-линейной аппроксимацией цветовых каналов на элементах сетки, вычислением разностного сигнала;
- кодированием разностного сигнала с применением квантуемого (для сжатия с потерями) дискретного косинусного или вейвлет-преобразования (предполагается использовать аналог *shape-adaptive* DCT или DWT алгоритма [18] для кодирования в каждом элементе сетки);
- применением энтропийного кодирования данных (сжатие без потерь, возможно применение LZMA-подобных методов сжатия, комбинаторных методов или иных методов энтропийного кодирования).

Необходимо указать некоторые особенности аппроксимации дискретных изображений с использованием треугольных сеток.

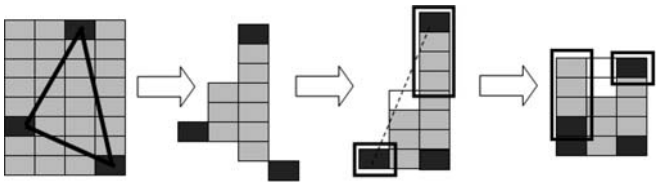


Рис. 1. Треугольный элемент сетки

Проблема принадлежности пикселей, лежащих на границе элементов сетки. Рассмотрим элемент сетки (рис. 1), наложенный на матрицу пикселей изображения. Треугольный элемент сетки наложен на пиксели изображения (вершины отмечены темным цветом); показаны трансформированный в прямоугольный треугольник и преобразованный в прямоугольник (искусственно синтезированные пиксели показаны белым).

Принадлежность пикселей элементу устанавливается с помощью перехода к симплексным координатам. Очевидно, что если пара узлов лежит на контуре (предположим, что контур может быть аппроксимирован прямой), ряд пикселей, принадлежащих "естественной" границе объектов изображения, будет лежать в этом треугольнике, а часть — в соседнем. Таким образом, линейная аппроксимация по вершинам треугольника даст существенные искажения в обоих треугольниках. Для определения коэффициентов аппроксимации можно применять точки, лежащие внутри треугольника. Линейная аппроксимация (с учетом указанной особенности) не позволяет достичь непрерывности цветового поля на границах треугольных элементов, что приводит к появлению в разностном сигнале высокочастотных составляющих.

Для вычисления дискретного косинусного преобразования (ДКП) невозможно использовать симплексные координаты. Дискретные значения координат пикселей, очевидно, не соответствуют дискретным значениям симплексных координат. Экспериментальная проверка показала, что использование интерполяции для перехода к равномерной сетке в симплексных координатах не эффективно.

Перенумерация пикселей. На первом этапе треугольник общего вида преобразуется в прямоугольный треугольник (рис. 1). Заметим, что на гипотенузе преобразованного треугольника имеется незаполненный пиксель, для которого нет соответствия в исходном элементе. Очевидно, имеется и противоположная возможность — появление пикселей, лежащих за границей преобразованного треугольника. Таким образом, любые дальнейшие преобразования принципиально не могут использовать симметрию матрицы пикселей (если, выполнив очевидные преобразования, считать, что преобразованному треугольнику соответствует верхнетреугольная матрица в идеальном случае). Это, в свою очередь, не позволяет использовать естественные свойства спектров симметричных сигналов, проводя,

например, децимацию матрицы коэффициентов преобразования.

Преобразования треугольных элементов к прямоугольным, основанные на перенумерации пикселей (с добавлением незначительного числа искусственно синтезированных пикселей), выполняемые, например, как показано на рис. 1, приводят к появлению высокочастотных составляющих спектра.

Насыщение высокочастотной части ДКП. Указанные выше особенности приводят к увеличению высокочастотной части ДКП-спектра, что не позволяет эффективно применять частотно-зависимое квантование коэффициентов ДКП (как, например, используется в стандарте JPEG). Эксперименты с различными матрицами квантования (синтезируемыми для треугольных элементов в симплексных координатах) показали, что соотношение качества и коэффициента сжатия не меняется кардинально. Также было получено, что неравномерное квантование не дает заметного выигрыша.

С точки зрения *вычислительной сложности* наиболее трудоемким является этап выделения контуров, поскольку неоднократно применяются операции цифровой фильтрации ко всему изображению. Ограничение размера конечных элементов и предразмещение узлов сетки позволяют достигать высокой скорости создания сетки.

Анализ эффективности алгоритма

Результат обработки изображения 2 показан на рис. 2. Результаты сжатия изображений приведены в табл. 2.

Сравнение эффективности предлагаемого подхода и алгоритма JPEG при сжатии изображений свидетельствует о том, что предлагаемый подход имеет превосходные характеристики в области высоких значений ПОСШ (при малых коэффициентах сжатия), хотя при весьма высоких коэффициентах сжатия кусочно-линейная аппроксимация на треугольной адаптивной сетке не позволяет получить приемлемый ПОСШ, при сохранении, однако, субъективного качества изображения (рис. 3). Ограничение коэффициента сжатия при больших шагах квантования ДКП определяется необходимостью хранения сетки и коэффициентов линейной аппроксимации.

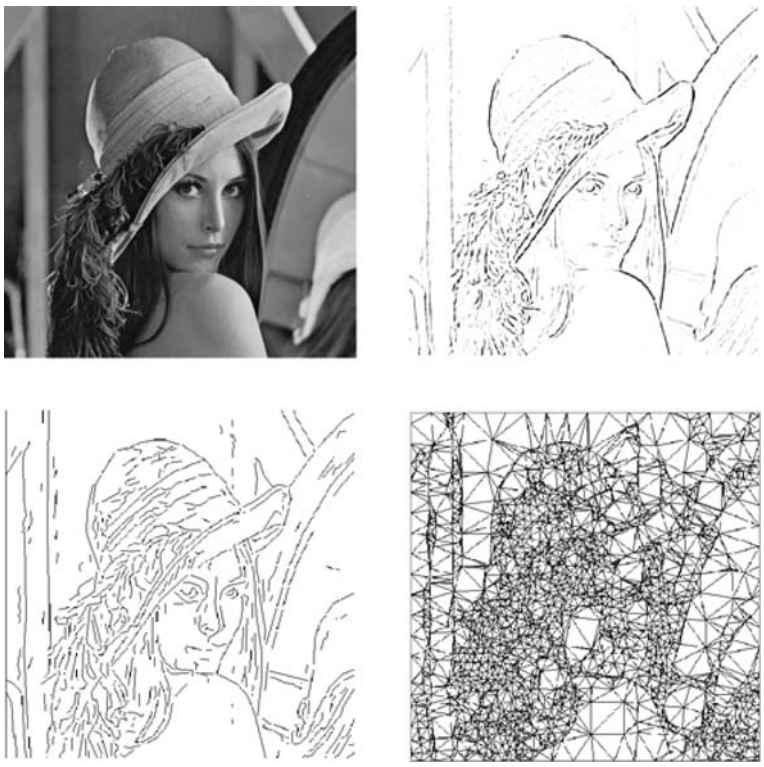


Рис. 2. Изображение 2 — исходное, обработанное, контуры, сетка

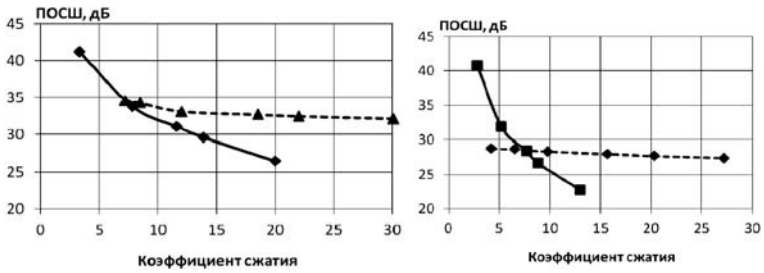


Рис. 3. Сравнение предлагаемого метода (сплошная линия) и JPEG (штриховая линия) при сжатии с потерями (слева — изображение 2, справа — изображение 3)

Таблица. 2

Коэффициент сжатия (числитель) и ПОСШ, дБ (знаменатель) для тестовых изображений

Изображение	ЛА	ЛА+ДКП 1	ЛА+ДКП 5	ЛА+ДКП 10	ЛА+ДКП 15
1	21,3/26,7	5,6/43,8	11,2/37,5	14,4/33,9	16,4/31,5
2	20,0/26,5	3,3/41,2	7,8/33,8	11,6/31,1	13,9/29,6
3	13,0/22,8	2,8/40,8	5,2/31,9	7,7/28,4	8,8/26,7
4	9,2/24,4	3,5/41,8	6,0/34,8	7,2/30,7	7,9/28,6

Примечание: ЛА — линейная аппроксимация; ДКП Q — применение ДКП с частотно-зависимым квантованием; ПОСШ — отношение пикового сигнала к шуму.

Заключение

Следует считать, что цель работы — эффективное построение адаптивной сетки изображения — достигнута, что доказывается сравнительно высоким значением ПОСШ при малом числе треугольных элементов сетки (см. табл. 1 и 2). При сжатии изображений выигрыша по сравнению с алгоритмом JPEG удается достигнуть при высоком ПОСШ (с учетом высокочастотного характера спектра раз-

ностного сигнала). В последующих статьях будут описаны результаты обнаружения движения объектов в видеосигналах на основе применения сеточных методов, а также результаты сжатия видеосигналов.

Работа выполнена при поддержке гранта РФФИ 12-07-00762.

Список литературы

1. **Ramirez-Agundis A., Gadea-Girones R., Colom-Palero R., Diaz-Carmona J.** A wavelet-VQ system for real-time video compression// Real-Time Image Proc-2007. Vol. 2. P. 271—280.

2. **Чобану М. К., Дворкович В. Н.** Проблемы и перспективы развития систем кодирования динамических изображений. Ч. 1—6 // MediaVision. 2011. № 2—8.

3. **Wiegand T., Sullivan G., Bjontegaard G.** et al. Overview of the H.264/AVC Video Coding Standard // IEEE Transactions on Circuits and Systems for Video Technology. 2011. N 13(7).

4. **Gao Y., Chen J., Yu S., Yang J., Zhou J.** A hybrid M-channel filter bank and DCT framework for H.264/AVC intra coding // Multimedia Tools Applications. 2009. 14 p.

5. **Robert A., Amonou I., Pesquet-Popescu B.** Improving DCT-Based Coders Through Block Oriented Transforms // ACIVS. 2006. P. 375—383.

6. **Mersereau R. M.** The processing of Hexagonally Sampled Two-Dimensional Signals // Proceedings of the IEEE. 1979. P. 930—949.

7. **Roussos A., Maragos P.** Reversible Interpolation of Vectorial Images by an Anisotropic Diffusion-Projection PDE // Int. Journal on Computer Vision. 2009. N 84. P. 130—145.

8. **Chun K.** et al. Method for irregular triangle mesh representation of an image based on adaptive control point removal // United States Patent 5923777, 13.07.1999.

9. **Liu W., Shao X. J., Dai C. H., He H. R.** Coupling mesh morphing and parametric shape optimization using SIMUOPTI // 3rd ANSA & μETA International Conference. 2009. 7 p.

10. **Вишняков С. В.** Применение вейвлет-преобразования для оптимизации сеток при расчете электромагнитных полей методом конечных элементов // Изв. РАН. Энергетика. 2010. № 6. С. 40—45.

11. **Чобану М. К.** Многомерные многоскоростные системы обработки сигналов. М.: Техносфера, 2009. 480 с.

12. **Lukin V. V., Fevralev D. V., Abramov S. K., Peltonen S., Astola J.** Adaptive DCT-based 1-D filtering of Poisson and mixed Poisson and impulsive noise // Proceedings of LNLA. Switzerland. 2008. P. 1—9.

13. **Lukin V., Krivenko S., Zriakhov M., Ponomarenko N., Abramov S., Kaarna A., Egiazarian K.** Lossy compression of images corrupted by mixed Poisson and additive noise // Proc. of LNLA, Helsinki. 2009. P. 33—40.

14. **Lukin V.** Processing of Multichannel RS data for Environment Monitoring // Proc. of NATO Advanced Research Workshop on Geographical Information Processing and Visual Analytics for Environmental Security, Italy. 2009. P. 129—138.

15. **Canny J.** A Computational Approach to Edge Detection // IEEE trans. on pattern analysis and machine intelligence. 1986. N 6. P. 679—698.

16. **Скворцов А. В.** Алгоритмы построения триангуляции с ограничениями // Вычислительные методы и программирование. 2002. Т. 3. С. 82—92.

17. **Robinson J.** Basic and Shape Sensivity Tests for Membrane and Plate Bending Finite Elements // Robinson and Associates. 1985.

18. **Foi A., Katkovnik V., Egiazarian K.** Pointwise Shape-Adaptive DCT for High-Quality Denoising and Deblocking of Grayscale and Color Images // IEEE transactions on image processing. 2007. Vol. 16. N 5. P. 1—17.

УДК 621.391

С. В. Дворников,
д-р техн. наук доц., проф. каф.,
e-mail: practicdsv@yandex.ru,
Д. В. Степынин,
канд. воен. наук, зам. начальника каф.,
А. С. Дворников, адъюнкт,
А. П. Букарева, курсант,
Военная академия связи, г. Санкт-Петербург

Формирование векторов признаков сигналов из вейвлет-коэффициентов их фреймовых преобразований

Представлены материалы исследований по распознаванию сигналов с близкой частотно-временной структурой. Предложено использовать в качестве признаков распознавания упорядоченные последовательности их вейвлет-коэффициентов. Обосновывается целесообразность синтеза вейвлет-коэффициентов на основе фреймовых преобразований. Анализируются результаты компьютерного моделирования.

Ключевые слова: *распознавание сигналов, вектор признаков, фреймовое преобразование*

Введение

В системе радиоконтроля вопросам распознавания вида модуляционного формата сигналов (в дальнейшем — распознавание сигналов) всегда уделялось особое внимание. Это связано с тем, что от эффективности их решения во многом зависит качество последующей идентификации источников радиоизлучений.

В настоящее время разработано множество научных подходов, позволяющих получить приемлемые результаты распознавания в условиях относительно низкого отношения сигнал/шум (ОСШ). Между тем специфика функционирования систем радиоконтроля, ориентированная на работу с кратковременными фрагментами излучений, существенно затрудняет их продуктивное использование [1].

Таким образом, указанная проблематика все еще является актуальной, а ее решение — значимым для систем радиоконтроля.

Общая постановка задачи распознавания сигналов в условиях обработки их кратковременных фрагментов реализаций

В общем случае задачу распознавания сигналов в системах радиоконтроля можно трактовать с позиций теории распознавания образов [2]. Тогда,

если в качестве объекта исследования определить реализацию обрабатываемого сигнала $\{\mathbf{Y}k\}$, где $k = 1, 2, ..., K$, то задачу распознавания можно трактовать как отнесение указанной реализации к одному из взаимоисключающих классов $\{\mathbf{S}\} = \{S_1, S_2, ..., S_L\}$, где $l = 1, 2, ..., L$, которые определяются совокупностью эталонных образов сигналов $\{\mathbf{S}_l\}$ с различными видами модуляционных форматов. Заметим, что $L \leqslant K$, т. е. число эталонов L может быть меньше числа распознаваемых классов сигналов K (в этом случае один из эталонов представляет собой множество всех нераспознанных классов сигналов, к которым может быть отнесена и входная реализация шума).

В свою очередь, каждая реализация $\{\mathbf{Y}k\}$ определяется совокупностью кратковременных фрагментов Yk^1, Yk^2, ..., Yk^P, являющихся конечным числовым множеством $\{\mathbf{Y}k\}$, где P — число фрагментов одной реализации. При этом каждая реализация отображается на множество K-классов распознаваемых модуляционных форматов $\{\mathbf{S}\} = \{S_1, S_2, ..., S_L\}$. Причем указанное отображение является однозначным, когда любая реализация $\{\mathbf{Y}k\}$ всегда определяется одним из значений $\{\mathbf{S}\}$, т. е. $\{\mathbf{S}_l\} \leftarrow \{\mathbf{Y}k\}$.

Заметим, что каждый из P фрагментов можно описать N признаками, в качестве которых могут выступать устойчивые параметры, на их основании можно сделать однозначное отнесение распознаваемого сигнала к одному из $\{\mathbf{S}\}$ взаимоисключающих классов.

Тогда каждый сигнал можно представить функцией вида

$$\{\mathbf{Y}k\} = d(Yk^1_{1...N}, Yk^2_{1...N}, ..., Yk^p_{1...N} ..., Yk^P_{1...N}), \quad (1)$$

где $n = 1, 2, ..., N$ — число признаков распознавания каждого из P фрагментов.

Таким образом, каждая реализация сигнала $\{\mathbf{Y}k\}$ представляет собой результат усреднения признаков его $\{\mathbf{X}k_n\}$ по всем кратковременным P фрагментам $Yk^1_{1...N}$, $Yk^2_{1...N}$, ..., $Yk^p_{1...N}$, ..., $Yk^P_{1...N}$.

Первоначальный набор признаков формируется из числа доступных измерению параметров сигнала, отражающих его наиболее существенные для распознавания свойства. Причем признаки выбираются таким образом, чтобы в результате их анализа обеспечивалась возможность различия характеризуемых ими классов, т. е. вектор разности признаков $\{\mathbf{X}kl\}$ был максимальным для любых k и l, принадлежащих множеству распознаваемых сигналов $\{\mathbf{Y}k\}$ и классов эталонных сигналов $\{\mathbf{S}\}$ [3].

Тогда процесс распознавания можно будет свести к процедурам вычисления вектора

$$\{\mathbf{X}kl\} = |\{\mathbf{Y}k\} - \{\mathbf{S}l\}|. \quad (2)$$

По своей сущности, вектор $\{\mathbf{X}kl\}$ представляет собой разность соответствующих признаков, характеризующих множества $\{\mathbf{Y}k\}$ и $\{\mathbf{S}l\}$. Очевидно, что корректность вычисления (2) подразумевает идентичность размерности признакового пространства для всех характеризуемых классов.

В работе [4] предлагается для оптимизации признаковых пространств $\{\mathbf{X}kl\}$ максимизировать функции $\{S_{1...N}\}$ по параметру n. Считается, что чем больше расстояние между векторами в признаковом пространстве, тем более описываемые ими классы контрастны [5]. Однако такой подход неизбежно приведет к увеличению значения N, что является нежелательным фактором. Поскольку увеличение признаковых пространств связано с усложнением системы распознавания. Поэтому задачу распознавания, применительно к рассматриваемой проблематике, сформулируем в следующем виде:

$$\max|\{\mathbf{Y}k\} - \{\mathbf{S}l\}|, \text{ для } \forall k \in \{\mathbf{Y}k_{1...N}\},\ l \in \{\mathbf{S}l_{1...N}\};$$
$$\text{при } N \to \min, \quad (3)$$

т. е. определить признаковое пространство, в котором минимальный набор признаков по N обеспечил бы максимальную контрастность распознаваемых объектов $\{\mathbf{Y}k\}$ или позволил получить максимальное значение для вектора $\{\mathbf{X}kl\}$, характеризующего межклассовое расстояние.

Формирование признаковых пространств на основе вейвлет-коэффициентов фреймовых преобразований

Анализ научных подходов к формированию признаковых пространств показал, что наибольшую контрастность при наименьшей размерности обеспечивают вейвлет-коэффициенты (ВК), синтезированные на основе различных масштабно-временных преобразований. Их эффективность определяется приемлемой степенью детализации распознаваемых сигналов, устойчивостью к аддитивным шумам и отсутствием в матрицах распределений ложных пиков энергии, обусловленных межсимвольной интерференцией [1, 3, 6].

В частности, в работе [6] обосновано использование фреймовых форм, поскольку они в большей степени приспособлены к анализу сигналов дискретной манипуляции.

Как правило, синтез фреймовых форм осуществляют на основе непрерывного вейвлет-преобразования, которое согласно работе [7] можно интерпретировать как скалярное произведение анализируемого сигнала $s(t)$ и базисных вейвлет-функций:

$$W(a, b) = \int_{-\infty}^{\infty} s(t) a^{-1/2} \psi^*\left(\frac{t-b}{a}\right) dt =$$
$$= \int_{-\infty}^{\infty} s(t) \psi^*_{a,b}(t) dt, \quad (4)$$

где $\psi(t)$ — материнский вейвлет, из которого с помощью процедур частотного масштабирования

и временных сдвигов формируются все базисные функции:

$$\psi_{a,\,b}(t) = a^{-1/2}\psi^*\left(\frac{t-b}{a}\right);\ a,\ b \in R,\ \psi \in L^2(R), \quad (5)$$

здесь параметр a определяет значение частотного масштаба и является аналогом частоты в анализе Фурье; параметр b определяет значение сдвига масштабированной вейвлет-функции вдоль оси времени.

В работе [7] показано, что значения параметра a обеспечивают возможность регулирования частотного разрешения. Следовательно, непрерывное вейвлет-преобразование автоматически обеспечивает возможность адаптивного изменения параметров функции-окна, в пределах которого происходит синтез матрицы масштабно-временного распределения энергии в зависимости от частотного наполнения обрабатываемого сигнала.

Между тем информационная избыточность матрицы распределения, синтезированной на основе (4), способствовала поиску новых форм представления на основе вейвлетов. Одной из которых и является фреймовое преобразование, реализуемое согласно работе [8] посредством полосовой фильтрации через гребенку фильтров, амплитудно-частотные характеристики которых определяются формой масштабирующих вейвлетов. В результате получают матрицу распределения, коэффициенты которой непрерывно изменяются вдоль оси времени и дискретно вдоль масштаба частоты, т. е. параметр a принимает дискретные значения, параметр b — непрерывные значения.

Фреймовое преобразование позволяет существенно снизить избыточность описания сигнала без потери информативности. Его использование для решения вопросов распознавания в работе [8] показало свою эффективность. Вместе с тем анализ применимости указанного технического решения к распознаванию сигналов со сложной частотно-временной структурой, к числу которых следует отнести восьмипозиционные сигналы фазовой манипуляции (ФМ-8), показал следующее. При обработке кратковременных фрагментов система распознавания на основе фреймовых преобразований не обеспечивает требуемую вероятность идентификации. Это связано с тем, что информационная компонента сигнала, учесть которую при формировании практически невозможно, вносит существенные изменения в вектор признаков.

В связи с этим предлагаются следующие основные этапы алгоритма формирования признаков и распознавания сигналов со сложной частотно-временной структурой.

1. Предварительно задают $L \geqslant 2$ эталонных сигналов $\{\mathbf{S}_l\}$, которые приводят к цифровой форме, для чего их дискретизируют и квантуют.

2. Для каждого l-го эталонного сигнала, где $l = 1, \ldots, L$, формируют матрицу распределения энергии $\mathbf{M}_l$ на основе фреймового преобразования. С этой целью отчеты оцифрованных сигналов пропускают через $M \geqslant 2$ фильтров, организованных следующим образом. Полосу пропускания $\Delta\Phi_m$ m-го фильтра, где $m = 1, \ldots, M$, выбирают из условия $\Delta\Phi_m = 2^{(m-1)}\Delta\Phi$, где $\Delta\Phi$ — ширина полосы пропускания первого фильтра.

3. Из ВК l-го эталонного сигнала, полученных с выхода каждой k-й полосы частот $\Delta\Phi_m$, формируют вектор признаков.

4. Распознаваемый сигнал $\{\mathbf{Y}k\}$ приводят к цифровой форме в соответствии с п. 1.

5. В соответствии с выражением (1) формируют фрагменты распознаваемого сигнала.

6. Для каждого фрагмента распознаваемого сигнала $\{\mathbf{Y}k\}$ формируют векторы признаков. Для этого выполняют операции, аналогично описанным в п. 2.

7. Формируют вектор признаков распознаваемого сигнала $\{\mathbf{Y}k\}$ путем усреднения векторов признаков всех P фрагментов.

8. Идентифицируют распознаваемый сигнал путем вычитания по модулю из его вектора признаков значения векторов признаков каждого эталонного сигнала в соответствии с выражением (2).

В качестве примера рассмотрим распознавание сигналов ФМ-8 в соответствии с разработанным алгоритмом. В качестве альтернативного излучения выбран близкий по структуре сигнал двойной фазовой манипуляции (ФМ-2) (рис. 1).

На рис. 2 показаны матрицы эталонных сигналов ФМ-8 и ФМ-2, синтезированных на основе фреймовых преобразований.

Векторы признаков (рис. 3) формируются путем построчной конкатенации строк матриц распреде-

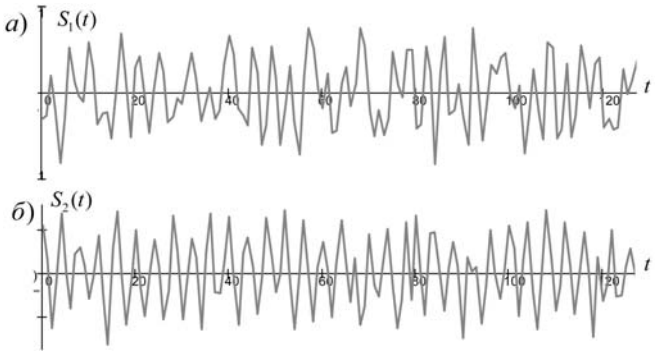


Рис. 1. Временное представление эталонных сигналов:
а — ФМ-8; *б* — ФМ-2

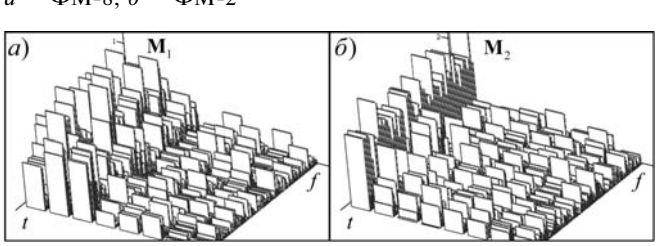


Рис. 2. Матрицы фреймового преобразования эталонных сигналов:
а — ФМ-8; *б* — ФМ-2

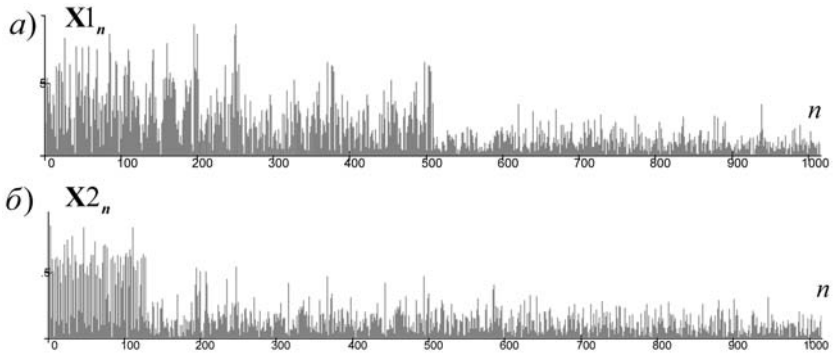


Рис. 3. Векторы признаков эталонных сигналов:
а — ФМ-8; *б* — ФМ-2

ления энергии. Причем для фреймовых преобразований число строк определяется значением масштабного коэффициента.

Очевидно, что значения ВК во многом определяются информационной составляющей обрабатываемого сигнала. Для снижения ее влияния на вектор признаков в работе [3] предлагается ВК ранжировать. Действительно, при рандомизации битовой последовательности указанная операция существенно снижает дисперсию ВК, составляющих результирующий вектор признаков.

Анализ ВК, составляющих вектор признаков, проведенный в работе [6], показал, что высоким значениям параметра масштабирования соответствуют коэффициенты шумов входной реализации. Как правило, они равномерно распределены в высокочастотной части матрицы и имеют амплитудные величины, уступающие сигнальным в абсолютном значении. Следовательно, без потери общности, они, как малозначимые, могут быть удалены из вектора признаков.

Формирование пространства векторов межклассового расстояния на основе выражения (2) позволяет получить совокупность значений {**X***kl*}, на основе которых принимается решение об отнесении распознаваемого сигнала к одному из альтернативных классов. В качестве примера на рис. 4 показаны вектор разности между распознаваемым сигналом ФМ-8 и эталонным сигналом ФМ-8 {**X**11}, а также эталонным сигналом ФМ-2 {**X**12}.

Следует заметить, что зависимость вектора **X***kl* не является линейной функцией от значения ОСШ. Это связано с тем, что шумовые составляющие локализованы в высокочастотной части матрицы фреймового представления сигнала, ВК которой не входят в состав усеченного вектора признаков.

В таблице представлены значения показателя $R = X12/X11$ в зависимости от ОСШ.

Следует отметить, что результаты, представленные в таблице, отображают близость ВК распознаваемого сигнала, в качестве которого использовался сигнал ФМ-8, коэффициентам первого эталона, также сформированного на основе сигнала ФМ-8.

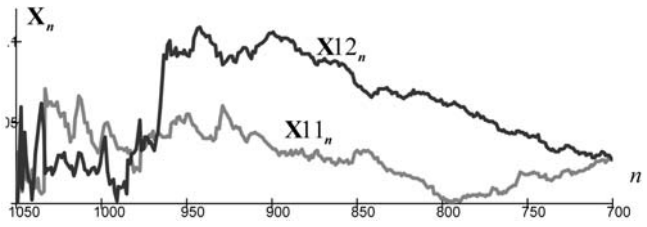


Рис. 4. Векторы разностей признаков значимых коэффициентов между: X12 — распознаваемым и эталонным сигналом ФМ-2; X11 — распознаваемым и эталонным сигналом ФМ-8

Зависимость показателя от *R* ОСШ

ОСШ, дБ	1	2	3	4	5	6
R	1,2	2,4	3,2	3,8	4,4	5,8

Заключение

Очевидно, что увеличение числа распознаваемых классов приведет к численному снижению показателя *R*. К тому же в практических приложениях целесообразно учитывать тот факт, что принимаемая реализация может содержать только шумы. Указанные вопросы требуют дальнейшего исследования.

Кроме того, следует заметить, что представленный подход может быть усовершенствован за счет уточнения допустимых границ, обусловленных дисперсией вектора признаков, вызванных информационной компонентой сигнала.

Между тем проведенные эксперименты подтвердили возможность распознавания сигналов с близкой частотно-временной структурой. Стоит отметить, что помехоустойчивость ВК фреймовых преобразований делает представленный подход весьма интересным для систем радиоконтроля.

Список литературы

1. **Дворников С. В., Дворников С. С., Коноплев М. А.** Алгоритм распознавания сигналов радиосвязи на основе симметрических матриц // Информационные технологии. 2010. № 9. С. 75—77.

2. **Фукунага К.** Введение в статистическую теорию распознавания образов: пер. с англ. М.: Наука, 1979. 368 с.

3. **Дворников С. В.** Метод формирования признаков сигналов для систем контроля и диагностики многоканальных систем радиосвязи // Контроль. Диагностика. 2009. № 5. С. 54—60.

4. **Ту Дж., Гонсалес Р.** Принципы распознавания образов / Пер. с англ. под ред. Ю. И. Журавлева. М.: Мир, 1978.

5. **Фомин Я. А., Тарловский Г. Р.** Статистическая теория распознавания образов. М.: Радио и связь, 1986. 272 с.

6. **Дворников А. С., Дворников С. В.** и др. Формирование признаков и распознавание сигналов на основе обработки их фреймовых преобразований // Информация и космос. 2011. № 2. С. 7—158.

7. **Дьяконов В. П.** Вейвлеты. От теории к практике. М.: СОЛОН-Р, 2002. 448 с.

8. **Способ** распознавания радиосигналов. Патент РФ № 2356064 МПК[7] G 06 K 9/00 от 20.05.2009 г.

УДК 519.72:621.372

В. А. Грушин, канд. техн. наук, доц.,
e-mail: grushin@lunn.ru,
Я. А. Архипова, аспирант,
e-mail: yand5833@yandex.ru,
Нижегородский государственный
лингвистический университет
им. Н. А. Добролюбова

Динамика экономического развития стран мира по фондовым индексам

Проводится анализ динамики развития экономики США, Великобритании, Японии и России с 1998 по 2011 гг. по ведущим биржевым индексам на основе нового информационного подхода к анализу случайных процессов. Приводятся таблицы информационного рассогласования индексов по годам и между собой с соответствующими выводами по динамике экономического развития. Приводится прогноз дальнейшей динамики индексов до 2014 г. включительно.

Ключевые слова: *динамика экономического развития, биржевые индексы, авторегрессионная модель наблюдений, метод обеляющего фильтра, информационное рассогласование, прогнозирование*

Введение

Существует множество методов анализа мировой экономики последнего периода новейшей истории, характеризующегося ее нестабильностью и кризисными явлениями. Несмотря на свою развитость классические методы и модели все менее эффективно работают в условиях глобализации мирового финансового пространства и расширения рынков. Экономические данные, в целом, носят стохастический характер, определяющее влияние на которые оказывает нестационарность социально-экономических процессов в обществе, их априорная неопределенность наряду с проблемой малых выборок, затрудняют проведение экономического анализа. Многие реальные процессы не могут адекватно быть описаны с помощью традиционных статистических моделей, поскольку являются существенно нелинейными и имеют хаотическую, квазипериодическую или смешанную основу [1]. Число факторов, определяющих их динамику, измеряется десятками и сотнями тысяч, при этом взаимосвязей между факторами неизмеримо больше. К важным особенностям задач анализа экономических процессов следует отнести также наличие и существенное влияние на них качественных факторов, что обусловливает трудность математической формализации модели [2].

Применяемые методы, такие как метод теории вероятностей, математической статистики, эконометрики, не дают эффективного результата. В последнее время разработаны новые нетрадиционные методы анализа стохастических процессов на базе авторегрессионной модели наблюдений, основанные на математической модели обеляющего фильтра, позволяющие классифицировать или объединять в группы (кластеры) сходные периоды экономического и общественного развития стран, отраслей производства и т. п. В частности, предлагается использовать информационный критерий для сравнения параметров случайных процессов (СП) с применением метрики Кульбака—Лейблера и анализа информационных параметров СП в виде их информационного рассогласования. Основой при этом служит разработанный проф. В. В. Савченко критерий минимума информационного рассогласования (МИР) [3]. Динамика экономического развития стран находит свое количественное отражение в биржевых экономических индексах.

Математическая модель экономического процесса

Общий подход к постановке задачи анализа динамики экономического процесса приводит к выбору классической модели временного ряда наблюдений, состоящей в разделении данных на две составляющие: детерминированную, включающую общие статистические закономерности изменения, и случайную — в виде "белого" шума. Наличие на входе модели ряда известных ретроспективных данных приводит к выбору авторегрессионной модели объекта. Хорошо разработанный математический аппарат и универсальность применения линейных методов анализа позволяет выполнить поиск линейной математической модели наблюдений. В итоге приходим к известной в теории статистических методов линейной стохастической модели временного ряда данных типа "авторегрессия" [4]

$$x_{t+1} = \sum_{i=1}^{q} a_i x_{t+1-i} + \eta_t,\ t = 1, 2, \ldots,$$

с порождающим "белым" шумом η_t в роли случайного возмущения с нулевым математическим ожиданием $M(\eta_t) = 0$ и некоторой фиксированной дисперсией $\sigma^2_{\eta_t}$. Вектор параметров a при этом состоит из соответствующих коэффициентов авторегрессии порядка q.

При сравнении однородности участков данных X_1 и X_2 биржевых индексов проверяются статистические гипотезы о разладке случайного процесса по конечным выборкам наблюдений в отношении автокорреляционных матриц (АКМ) K_1 и K_2 соответствующих выборок X_1 и X_2 [5]:

$$W_0 : K_1 = K_2 \triangleq K_0$$

против альтернативы

$$W_1 : K_1 \neq K_2.$$

Для частотной области обработки сигналов и при дополнительном условии нормировки авторегрессионные модели (АР-модели) процессов по дисперсии их порождающего шума выражение для решающей статистики принимает вид

$$W_1 : \lambda(X_0) =$$

$$= \frac{1}{F}\sum_{f=1}^{F}\frac{\left|1 + \sum\limits_{m=1}^{p} a_x(m)\exp(-j\pi mf/F)\right|^2}{\left|1 + \sum\limits_{m=1}^{p} a_1(m)\exp(-j\pi mf/F)\right|^2} - 1 > \tilde{\lambda}_0, \quad (1)$$

где $\{a_1(m)\}$, $\{a_x(m)\}$ — k-векторы коэффициентов линейной среднеквадратической авторегрессии сигналов $X_1(t)$ и $X_2(t)$ соответственно, полученных с применением метода обеляющего фильтра (МОФ); p — порядок фильтра; F — частота; m — порядок АР-модели; $\tilde{\lambda}_0$ — пороговый уровень [6].

Выражение (1) описывает выборочную оценку информационного рассогласования (ИР) между сигналом X на входе и опорным сигналом в частотной области [7]. Последовательно применяя выражение (1) к выделенным участкам данных, мы анализируем изменение величины ИР между исследуемыми индексами. Полученные данные помещаются в таблицу взаимных информационных рассогласований (ВИР).

Сводя решаемую задачу анализа динамики относительного экономического развития стран (регионов) к выбору оптимальной АР-модели, воспользуемся при анализе качества последней универсальным критерием минимума информационного рассогласования в метрике Кульбака—Лейблера [8]:

$$I_n[f_R|f_K] = \int\ldots\int \ln\left[\frac{f_K(x_n)}{f_R(x_n)}\right] f_K(x_n)dx_n \to \min_R.$$

Здесь $f_K(x_n)$, $f_R(x_n)$ — плотности n-мерных гауссовских распределений с автокорреляционными матрицами анализируемого процесса $K_{n,\,n}$ и его модели $R_{n,\,n}$ соответственно.

Исходя из энтропии анализируемого распределения, определяемого как

$$H^*_n(X) = \frac{1}{2}\ln(2\pi\mathbf{e})^n|K_{n,\,n}|] = \frac{n}{2}[\ln(2\pi\sigma_0^2)] + \frac{n}{2},$$

переходим к удельной (на один отсчет данных) величине информационного рассогласования:

$$I_n[f_R|f_K] = \frac{1}{n} I_n[f_R|f_K]_{n\to\infty} =$$

$$= \frac{1}{2}\left[\ln(2\pi\sigma_\eta^2) + \frac{\sigma_z^2(q)}{\sigma_\eta^2}\right] - h^*(X),$$

где $h^*(X) = \frac{1}{2}\ln(2\pi\sigma_0^2) + \frac{1}{2} = \text{const}$ — удельная энтропия случайного процесса.

Отсюда следует информационный критерий оптимальности АР-модели наблюдений вида

$$\Delta h(q,\,n) = \ln\sigma_\eta^2(q,\,n) + \frac{\sigma_z^2(q,\,n)}{\sigma_\eta^2(q,\,n)} \to \min.$$

Здесь $\sigma_\eta^2(q,\,n)$, $\sigma_z^2(q,\,n)$ — зависимости дисперсий порождающего шума и ошибки прогнозирования от двух исходных параметров адаптивной оценки: порядка q и объема анализируемой (обучающей) выборки n. [9]

В условиях нормировки $\sigma_\eta^2(q,\,n) = 1$ получаем, что ИР зависит от отношения дисперсий (мощностей) порождающего шума на выходе обеляющего фильтра. При большом диапазоне сравниваемых величин применяется логарифмическая мера, позволяющая отношение величин заменить их разностью, что гораздо удобнее при обработке данных. Для сравнения мощностей в качестве логарифмической меры отношения пользуются выражением

$$10\lg\frac{P_1}{P_0} = 10\lg\frac{\sigma_z^2(q,\,n)}{\sigma_\eta^2(q,\,n)}.$$

Проведение экспериментальных исследований

В настоящее время в мире действует порядка 200 фондовых бирж. Крупнейшие — Токийская (Tokyo Stock Exchange, TSE), Нью-Йоркская (New York Stock Exchange, NYSE), Лондонская (London Stock Exchange, LSE), Московская Межбанковская валютная биржа (ММВБ) — могут считаться регулятором и индикатором состояния мировой экономики [3].

Рассмотрим динамику развития мировой экономики на примере информационного анализа и сравнения ведущих мировых биржевых индексов. За основу исследования взяты индексы FTSE 100 (LSE), S & P 500 (NYSE), ММВБ, Nikkei 225 (TSE)

в период с 1998 по 2011 гг. с сегментацией в один год. Длина анализируемого массива данных 252 отсчета. Порядок модели обеляющего фильтра $q = 30$ [3].

В табл. 1—4 представлены взаимные информационные рассогласования индексов, рассчитанные путем последовательного применения формулы (1).

Динамику ИР индексов рассмотрим графически относительно 2000 г.

Из представленного графика (рис. 1) видно, что в период с 1998 по 2011 гг. экономика Японии дважды переживала кризис — в 2003 и 2009—2011 гг. В 2003 г. наблюдался спад экономики до уровня −2,5 дБ. С 2003 по 2007 гг. на выходе из кризиса был достигнут уровень экономики 2000 г., после чего снова начался кризисный спад 2009— 2011 гг. Причем уровень 2011 г. ниже уровня 2009 г., что

Таблица 1

Информационное рассогласование индекса FTSE 100 по годам

	1998	1999	2000	2001	...	2008	2009	2010	2011
1998	0,0000	−0,4763	−0,5368	0,0693	...	0,2824	0,9789	0,1404	−0,0347
1999	0,4884	0,0000	−0,0593	0,5568	...	0,7804	1,4381	0,6134	0,4508
2000	0,5563	0,0688	0,0000	0,6172	...	0,8324	1,5061	0,6775	0,5135
2001	−0,0438	−0,5212	−0,5894	0,0000	...	0,2087	0,9384	0,0921	−0,0939
2002	−0,8844	−1,3535	−1,4241	−0,8442	...	−0,6518	0,1269	−0,7424	−0,9372
2003	−1,4101	−1,9070	−1,9684	−1,3382	...	−1,1061	−0,4954	−1,2979	−1,4475
2004	−0,9348	−1,9070	−1,4897	−0,8717	...	−0,6439	0,0026	−0,8201	−0,9778
2005	−0,3572	−0,8506	−0,9136	−0,2874	...	−0,0579	0,5695	−0,2427	−0,3950
2006	0,2364	−0,2534	−0,3169	0,3019	...	0,5288	1,1774	0,3521	0,1951
2007	0,5753	0,0879	0,0239	0,6384	...	0,8575	1,5229	0,6986	0,5316
2008	−0,2043	−0,6733	−0,7474	−0,1634	...	0,0000	0,8059	−0,0556	−0,2556
2009	−0,8784	−1,3846	−1,4448	−0,8044	...	−0,5552	0,0000	−0,7812	−0,9160
2010	−0,1079	−0,5999	−0,6639	−0,0424	...	0,1866	0,8297	0,0000	−0,1507
2011	0,0494	−0,4299	−0,4956	0,1038	...	0,3150	1,0254	0,1817	0,0000

Таблица 2

Информационное рассогласование индекса ММВБ по годам

	1998	1999	2000	2001	...	2008	2009	2010	2011
1998	0,0000	−0,2203	−0,6173	−0,5658	...	−1,3960	−1,2578	−1,4582	−1,5164
1999	0,4020	0,0000	−0,3006	−0,2832	...	−1,0317	−1,0118	−1,1573	−1,1958
2000	0,6981	0,3678	0,0000	0,0491	...	−0,7564	−0,6639	−0,8361	−0,8860
2001	0,6631	0,3106	−0,0265	0,0000	...	−0,7688	−0,7161	−0,8791	−0,9218
2002	0,8615	0,5159	0,1705	0,2048	...	−0,5728	−0,5134	−0,6776	−0,7218
2003	1,0381	0,6556	0,3269	0,3537	...	−0,3985	−0,3743	−0,5283	−0,5649
2004	1,1390	0,8051	0,4480	0,4878	...	−0,2918	−0,2294	−0,3994	−0,4435
2005	1,2608	0,8878	0,5605	0,5828	...	−0,1585	−0,1447	−0,3014	−0,3347
2006	1,5262	1,1763	0,8340	0,8655	...	0,0984	0,1455	−0,0187	−0,0595
2007	1,6205	1,2796	0,9307	0,9637	...	0,1918	0,2469	0,0790	0,0367
2008	1,4821	1,2000	0,8091	0,8651	...	0,0000	0,1722	−0,0156	−0,0802
2009	1,4203	1,0292	0,7120	0,7349	...	−0,0044	0,0000	−0,1466	−0,1801
2010	1,5408	1,2025	0,8533	0,8859	...	0,1165	0,1688	0,0000	−0,0412
2011	1,5743	1,2586	0,8935	0,9348	...	0,1358	0,2269	0,0500	0,0000

Таблица 3

Информационное рассогласование индекса S&P 500 по годам

	1998	1999	2000	2001	...	2008	2009	2010	2011
1998	0,0000	−0,8759	−1,1773	−0,3858	...	−0,3359	0,6571	−0,2108	−0,6669
1999	0,8837	0,0000	−0,3035	0,4918	...	0,5400	1,5267	0,6650	0,2127
2000	1,2072	0,3212	0,0000	0,7999	...	0,8101	1,8562	0,9848	0,5238
2001	0,4326	−0,4500	−0,7679	0,0000	...	0,0216	1,1012	0,2118	−0,2627
2002	−0,3640	−1,2439	−1,5690	−0,8010	...	−0,8064	0,3339	−0,5922	−1,0704
2003	−0,4945	−1,3817	−1,6816	−0,8766	...	−0,8097	0,1172	−0,7163	−1,1567
2004	0,2094	−0,6750	−0,9829	−0,1935	...	−0,1483	0,8568	−0,0163	−0,4700
2005	0,4967	−0,3881	−0,6975	0,0960	...	0,1371	1,1394	0,2736	−0,1796
2006	0,8527	−0,0326	−0,3376	0,4571	...	0,5114	1,4902	0,6244	0,1779
2007	1,3653	0,4806	0,1686	0,9639	...	0,9959	2,0069	1,1430	0,6858
2008	0,5296	−0,3534	−0,6996	0,0788	...	0,0000	1,2276	0,3110	−0,1852
2009	−0,5851	−1,4751	−1,7694	−0,9580	...	−0,8704	0,0000	−0,8105	−1,2386
2010	0,2312	−0,6525	−0,9574	−0,1633	...	−0,1112	0,8755	0,0000	−0,4448
2011	0,6906	−0,1894	−0,5037	0,2777	...	0,2996	1,3615	0,4706	0,0000

Таблица 4

Информационное рассогласование индекса Nikkei 225 по годам

	1998	1999	2000	2001	...	2008	2009	2010	2011
1998	0,0000	−0,3554	−0,4512	1,0698	...	1,1636	2,2098	1,8579	2,1287
1999	0,4540	0,0000	0,0227	1,5418	...	1,6665	2,5592	2,2884	2,5848
2000	0,4749	0,1410	0,0000	1,5341	...	1,6112	2,7070	2,3277	2,5963
2001	−1,0415	−1,3793	−1,5037	0,0000	...	0,0890	1,1795	0,8112	1,0787
2002	−1,8122	−2,1670	−2,2747	−0,7605	...	−0,6852	0,3961	0,0407	0,3118
2003	−2,0989	−2,5554	−2,5317	−1,0006	...	−0,8751	−0,0074	−0,2666	0,0248
2004	−1,3467	−1,7472	−1,7970	−0,2750	...	−0,1700	−0,0074	0,4926	0,7780
2005	−0,8359	−1,2882	−1,2636	0,2614	...	0,4180	1,2688	0,9830	1,2936
2006	0,2451	−0,1549	−0,2064	1,3172	...	1,4373	2,4031	2,0772	2,3692
2007	0,4495	0,0719	−0,0066	1,5135	...	1,6039	2,6300	2,2990	2,5694
2008	−1,0533	−1,3725	−1,5279	−0,0132	...	0,0000	1,1780	0,8184	1,0632
2009	−2,0826	−2,5410	−2,5142	−0,9987	...	−0,8845	0,0000	−0,2497	0,0432
2010	−1,8318	−2,2107	−2,2870	−0,7665	...	−0,6497	0,3517	0,0000	0,2926
2011	−2,1174	−2,4685	−2,5763	−1,0559	...	−0,9669	0,0906	−0,2638	0,0000

говорит о кризисном застое экономики Японии в этот период.

Аналогичную картину наблюдаем и при анализе экономики США (рис. 1) за тот же период. Однако с 2009 г. экономика США пошла в рост и к 2012 г. достигла уровня 2000 г.

По данным FTSE 100 (рис. 1) в экономике Великобритании также видим два кризисных провала от уровня 2000 г. Причем мировой кризис 2003 г. больше повлиял на экономику страны, чем кризис 2009 г., и к настоящему времени также наблюдается процесс выхода из кризиса.

По данным ММВБ (рис. 1) экономика России развивалась довольно устойчиво, мировой кризис 2003 г. не сильно повлиял на рост экономики. А кризис 2009 г. незначительно повлиял на экономическую ситуацию в стране (−0,2 дБ), и в 2011 г. был практически достигнут уровень экономики 2007 г.

Рассмотрим динамику индексов ведущих мировых фондовых бирж по годам (табл. 5). Сравнение индексов проводится по МОФ в переломные моменты экономического развития. Из табл. 5 видно, что в исследуемый период самой сильной являлась экономика Японии. В 2000 г. самой слабой была экономика России, но к 2007 г. динамика экономики России приближается к уровню динамики экономики США. Страны Европы и Азии повторяют динамику

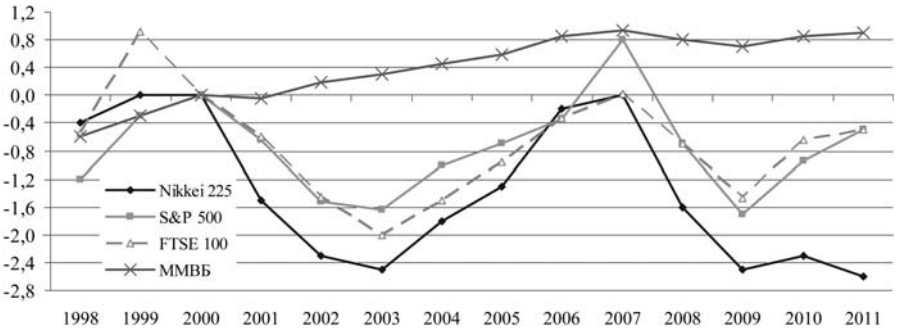


Рис. 1. Динамика информационных рассогласований индексов, дБ

Таблица 5

Сравнение динамики индексов по годам, дБ

2000 г.	S&P 500	Nikkei 225	FTSE 100	ММВБ
S&P 500	0,00000000	−1,07737200	−0,65386919	0,84206300
Nikkei 225	1,08230960	0,00000000	0,42844728	1,92117000
FTSE 100	0,65431480	−0,42299500	0,00000000	1,49715100
ММВБ	−0,83495800	−1,91533200	−1,48823094	0,00000000
2003 г.	S&P 500	Nikkei 225	FTSE 100	ММВБ
S&P 500	0,00000000	−0,98766900	−0,62633710	0,35914200
Nikkei 225	0,98921660	0,00000000	0,36302031	1,34719200
FTSE 100	0,62664220	−0,36087500	0,00000000	0,98643000
ММВБ	−0,35354700	−1,34256100	−0,97935397	0,00000000
2007 г.	S&P 500	Nikkei 225	FTSE 100	ММВБ
S&P 500	0,00000000	−1,06062300	−0,63967749	−0,06851000
Nikkei 225	1,06219790	0,00000000	0,42234506	0,99437300
FTSE 100	0,63983930	−0,42095100	0,00000000	0,57127700
ММВБ	0,06990530	−0,98992000	−0,56982808	0,00000000
2009 г.	S&P 500	Nikkei 225	FTSE 100	ММВБ
S&P 500	0,00000000	−1,00072600	−0,68764469	−0,02641000
Nikkei 225	1,00238760	0,00000000	0,31489094	0,97574300
FTSE 100	0,68810730	−0,31248300	0,00000000	0,66286600
ММВБ	0,03744450	−0,96309000	−0,64935396	0,00000000
2011 г.	S&P 500	Nikkei 225	FTSE 100	ММВБ
S&P 500	0,00000000	−0,87222300	−0,65490536	−0,10254000
Nikkei 225	0,87365270	0,00000000	0,21840216	0,77003000
FTSE 100	0,65509660	−0,21746700	0,00000000	0,55227700
ММВБ	0,01437140	−0,76890100	−0,55080402	0,00000000

развития экономики Америки в тех же пропорциях, что подтверждает глобализацию мировой экономики.

Обеляющий фильтр как прогнозная модель случайного процесса

Прогноз СП X_t на один шаг в будущее есть в общем случае функция имеющихся наблюдений $y_n = y(x_0, ..., x_{n-1})$. Из математической статистики известно, что оптимальная оценка прогнозирования определяется по формуле регрессии n-го (ненаблюдаемого) отсчета случайного процесса X_t относительно вектора его последовательных наблюдений:

$$y_n^* = M\{x_n | x_0, ..., x_{n-1}\}, \quad (2)$$

где M — символ математического ожидания.

Задавая в качестве модели формирования СП X_t модель авторегрессии, будем иметь зависимость

$$x_t = a_1 x_{t-1} + ... + a_q x_{t-q} + \eta_t;\ t = 1, 2, ..., \quad (3)$$

фиксированного порядка $q < \infty$ с порождающим "белым шумом" $\{\eta_t\}$; на входе будем иметь линейную зависимость оценки прогнозирования

$$y_n = a_1 x_{n-1} + ... + a_q x_{n-q} \quad (4)$$

того же порядка q, если $q < n$. Здесь $a_1, ..., a_q$ — q-вектор коэффициентов авторегрессии, которые не зависят от времени, т. е. являются постоянными величинами, если процесс X_t на всем интервале наблюдения $t < n$ стационарен в широком смысле. Условия такой стационарности обеспечиваются на практике введением жестких ограничений на объем наблюдений n. Дисперсия ошибки прогнозирования равна в рассматриваемом случае дисперсии или средней мощности порождающего шума. При дополнительном условии гауссовости процесса X_t этим достигается наивысшая или потенциально возможная точность прогнозирования. Иными словами, в предположении о гауссовости и стационарности анализируемого процесса оптимальный прогноз (2) сводится к линейной зависимости (4).

Зависимость (4) легко преобразуется в рекуррентное выражение для оптимального прогноза на произвольное число шагов $k + 1$ в будущее вида

$$y_{n+k} = a_1 y_{n+k-1} + ... + \\ + a_k y_k + a_{k+1} x_{n-1} + ... + a_q x_{n+k-q}, \quad (5)$$

где вместо ненаблюдаемых отсчетов $x_n, ..., x_{n+k-1}$ используются их предварительные оценки прогнозирования на меньшее число шагов до k-го включительно.

Таким образом, при оптимальном или, в ряде случаев, близком к оптимальному решении задачи прогнозирования в формулировке (2) выполняются следующие операции над данными наблюдений:

- определяется АР-модель наблюдений согласно выражению (3);
- с использованием ее параметров по формуле (4) рассчитывается прогноз случайного процесса X_t на один шаг;
- из выражения (5) по индукции при $k = 1, 2, ...$ дается требуемый прогноз на произвольное число шагов в будущее [7].

Прогнозы динамики ведущих мировых индексов представлены на рис. 2.

Согласно полученному прогнозу экономику России ждет дальнейшее преобладание положительной динамики (рис. 2, *а*). Небольшой рост экономического развития Великобритании в 2012 г. увеличится в 2013 г. на 0,6 дБ относительно 2011 г., с откатом в 2014 г. на уровень 2013 г. (рис. 2, *б*). Наблюдаем положительную динамику и в Японии до 2,5 дБ в 2014 г. относительно 2011 г. (рис. 2, *в*). Падение экономики США относительно 2011 г. на −0,3 дБ в 2012 г. сменится последующим ростом на 0,3...0,4 дБ в 2014 г. (рис. 2, *г*).

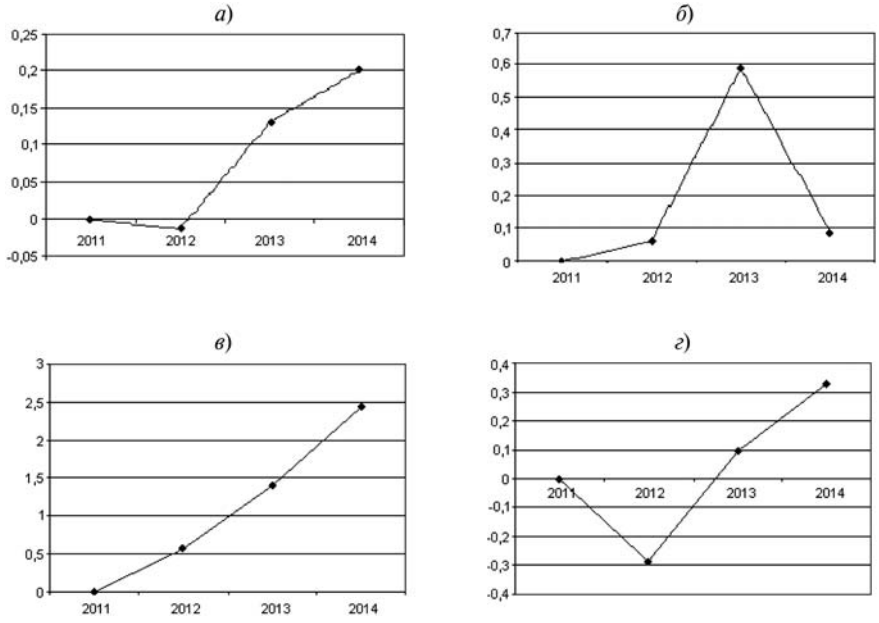


Рис. 2. Прогноз динамики ведущих мировых индексов ММВБ (*а*), FTSE 100 (*б*), Nikkei 225 (*в*), S & P 500 (*г*) на 2012—2014 гг., дБ

Прогнозы динамики индексов (рис. 2) носят качественный характер, так как сделаны по малой выборке, равной 14 годовым значениям ИР относительно 2011 г. Порядок модели $q = 13$.

Заключение

Применение новых информационных методов в исследовании динамики экономического развития стран по биржевым индексам позволило количественно оценить и

наглядно проследить данную динамику. За рассматриваемый период с 1998 по 2011 гг. произошло два кризиса мировой экономики с достижением при выходе уровня экономики 2000 г. Применяемый метод обеляющего фильтра позволяет корректно проводить сравнение динамики индексов — основных показателей экономик стран — между собой в силу введения нормировки математической модели СП данных по уровню порождающего шума. Прогнозирование дальнейшего развития индексов по их информационному рассогласованию позволяет определить качественный характер развития каждого индекса относительно другого и, соответственно, экономик, состояние которых данные индекс отражают.

Информационный подход имеет общий характер и может быть применен при анализе любых статистических данных, независимо от их экономической, общественной, социальной, физической и другой природы.

Список литературы

1. **Некипелов Н.** Опыт прогнозирования финансовых рынков // Технологии анализа данных: Сб. статей. URL: http://basegroup.ru/neural/stockmarket.html.
2. **Abraham B., Ledolter J.** Statistical methods for forecasting. New York: Willey, 2003.
3. **Акатьев Д. Ю., Савченко В. В.** Обнаружение разладки случайного процесса по выборке на основе принципа минимума информационного рассогласования // Автометрия. 2005. № 2. С. 68—74.
4. **Савченко В. В., Акатьев Д. Ю., Баринов А. В., Шкулев А. А.** Программное обеспечение биржевой игры на основе данных краткосрочного прогнозирования: Учеб. пособие / Под общей ред. В. В. Савченко. Н. Новгород: НГЛУ им. Н. А. Добролюбова, 1999. 87 с.
5. **Архипова Я. А., Грушин В. А.** Диагностика однородности случайного временного ряда биржевых котировок на основе критерия минимума информационного рассогласования // Проблемы управления в сложных экономических и социальных системах: Сб. научных статей. Вып. 2 / Под общей ред. В. А. Бородина. Н. Новгород: НГЛУ им. Н. А. Добролюбова, 2011. С. 20—25.
6. **Архипова Я. А., Грушин В. А.** Определение границ однородности участков данных социально-экономических процессов по методу обеляющего фильтра переменного порядка // Проблемы управления в сложных экономических и социальных системах: Сб. научных статей. Вып. 2 / Под общей ред. В. А. Бородина. Н. Новгород: НГЛУ им. Н. А. Добролюбова, 2011. С. 26—32.
7. **Савченко В. В.** Обнаружение и прогнозирование разладки случайного процесса на основе спектрального оценивания // Автометрия. 1996. № 2. С. 77—84.
8. **Савченко В. В.** Тестирование спектральных оценок по выборке // Автометрия. 1999. № 4. С. 107—112.
9. **Архипова Я. А.** Определение границ однородности участков данных (кластеров) по критерию минимума информационного рассогласования // Всероссийский конкурс научно-исследовательских работ студентов и аспирантов в области информатики и информационных технологий в рамках всероссийского фестиваля науки 7 сентября—9 сентября 2011. Сб. научных работ. Белгород: БелГУ, 2011. Т. 2. С. 320—328.
10. **Краус М., Вошни Э.** Измерительные информационные системы. М.: МИР, 1975.

Новые книги

Липаев В.В.

Очерки истории отечественной программной инженерии 1940-е — 80-е годы. — М.: СИНТЕГ, 2012. 245 с.

Описание отечественной программной инженерии начинается с появления в нашей стране электронных вычислительных машин (ЭВМ) и программирования в 1940-е — 60-е годы. Далее изложена история проектирования и производства отечественных ЭВМ, а также средств и систем автоматизации технологических процессов производства программных продуктов в 1960-е — 80-е годы. Подробно представлена история формирования основных компонентов программной инженерии в 1960-е — 70-е годы. Внимание акцентируется на особенностях решения сложных задач по государственным заказам и на создании программных продуктов для мобильных и бортовых ЭВМ реального времени. Особое внимание уделяется истории разработки методов моделирования динамических объектов и стендов для тестирования и испытаний комплексов программ в реальном времени. Изложены методы оценивания качества программных продуктов, рисков, дефектов и ошибок при их разработке, а также история формирования требований к профессиям и квалификации специалистов программной инженерии в 1970-е — 80-е годы. Проведен анализ сложности программных комплексов реального времени и распределение ресурсов ЭВМ для таких комплексов, характеристики и методы оценивания качества их компонентов. Один из разделов посвящен истории формирования в 1980-е годы экономики программной инженерии, созданию средств технико-экономический анализа и экономическому обоснованию планов разработки крупных программных продуктов. Представлены реальные примеры их создания в 1960-е — 80-е годы для оборонных систем на основе методов программной инженерии.

Книга предназначена для специалистов по вычислительной технике и программной инженерии, программистов, студентов и аспирантов, интересующихся историей развития, успехами и проблемами отечественной науки и техники в этой области.

УДК 004.032.26

С. Н. Данилин, канд. техн. наук,
М. В. Макаров, аспирант,
С. А. Щаников, аспирант,
Муромский институт (филиал)
федерального государственного бюджетного
образовательного учреждения
высшего профессионального образования
"Владимирский государственный университет
имени Столетовых",
e-mail: nauka-murom@yandex.ru

Комплексный показатель качества работы нейронных сетей

Предложен комплексный показатель определения качества (точности) работы нейронных сетей, учитывающий значения функциональных допусков. Приведено описание свойств комплексного показателя и его использования на примере нейронных сетей, реализующих аппроксимацию базовых математических функций. Показана зависимость качества работы нейронных сетей от выбранной функции обучения.

Ключевые слова: *нейронные сети, точность работы, качество работы нейронной сети*

При разработке алгоритмов преобразования информации в любом логическом базисе и проектировании устройств, реализующих данные алгоритмы, в соответствии с действующими российскими и международными стандартами устанавливаются технические требования к ним, в частности, по качеству (точности) работы, быстродействию, отказоустойчивости, надежности [1].

В процессе проектирования устройств с нейросетевой архитектурой или работающих в нейросетевом логическом базисе (нейронных сетей), в зависимости от решаемых задач, наиболее часто выбирают один из типовых показателей качества (точности) их работы: сумму квадратов ошибок (SSE), среднюю квадратическую ошибку (MSE), среднюю абсолютную ошибку (MAE), абсолютную $\sigma_{абс}$ или относительную $\sigma_{отн}$ ошибку (погрешность) решения задачи [2].

Общим недостатком перечисленных выше критериев является отсутствие учета в них допустимых уровней (допусков) качества работы нейронных сетей, нормируемых как отечественными, так и зарубежными стандартами при проектировании и функционировании любых технических объектов [3].

Предлагается комплексный показатель качества (точности) работы нейронных сетей K при следующих установленных значениях функциональных допусков:

$$K_i = 1 - (x_i - x_{дос})/(x_{доп} - x_{дос}), \text{ при } x_{доп} > x_{дос}; \quad (1)$$

$$K_i = 1 - (x_{дос} - x_i)/(x_{дос} - x_{доп}), \text{ при } x_{доп} < x_{дос}, \quad (2)$$

где $x_{доп}$ — допустимое значение (допуск) показателя качества работы нейронной сети; $x_{дос}$ — значение показателя качества работы нейронной сети, достигнутое при обучении; x_i — значение показателя качества работы нейронной сети, при вариации параметра i-го нейрона или элемента сети.

Для количественной оценки качества работы всей нейронной сети предложен обобщающий комплексный критерий — средний комплексный показатель качества $K_{ср}$, определяемый выражением

$$K_{ср} = \frac{1}{N}\sum_{i=1}^{N} K_i, \quad (3)$$

где N — число нейронов или структурных элементов в нейронной сети.

Критерий K является безразмерной величиной и может иметь значения от 1 до $-\infty$. Диапазон изменения K от 1 до 0 характеризует запас погрешности (показателя качества) технического объекта до достижения границы поля допуска. Если значение показателя качества работы нейронной сети при каких-либо изменениях параметров ее составных элементов становится ниже допустимого уровня (границы поля допуска), то нейронная сеть не является работоспособной. В этом случае критерий K становится отрицательным и его абсолютное значение, увеличенное на единицу, показывает во сколько раз изменение значения показателя качества работы нейронной сети при вариации параметра соответствующего элемента (нейрона) превышает допустимое изменение (границу поля допуска). Чем ближе значение коэффициента к 1, тем более высокое качество (точность) работы нейронной сети.

Исследования авторов показали [4, 5], что одна и та же нейронная сеть имеет различную восприимчивость (чувствительность) к вариациям параметров ее элементов при использовании разных критериев качества обучения и работы.

Исследуем в качестве примера с помощью критерия $K_{ср}$ нейронные сети, реализующие преобразование информации по следующим алгоритмам:

а) $y = \sin(\beta)$ при $\beta \in [0; \pi/2]$, шаг дискретизации входной информации 10^{-4}, допустимая относительная погрешность работы НС — не более 2 %;

б) $y = 1/\sqrt{x}$ при $x \in [1; 2]$, шаг дискретизации входной информации 10^{-4}, допустимая относительная погрешность работы НС — не более 2 %.

Для этого создадим двухслойные нейронные сети прямого распространения с шестью и восемью нейронами в первом слое (логистическая функция активации) и одним нейроном во втором (линейная функция активации).

Исследования проводят по следующей методике:

1. Каждая сеть обучается до достижения наилучшего результата по критерию (показателю качества) средней квадратичной ошибки (MSE) в соответствии со следующими алгоритмами:

а) TRAINLM — метод Левенберга—Маркара;

б) TRAINBR — функция тренировки на основе обратного распространения ошибки с использованием Байесовской регуляризации;

в) TRAINSCG — метод шкалированных связанных градиентов;

г) TRAINGD — метод градиентного спуска;

д) TRAINR — метод случайных приращений;

е) TRAINRP — алгоритм упругого обратного распространения.

2. Фиксируются значения максимальной абсолютной $E_{абс}$ и относительной $E_{отн}$ погрешностей, возникающих при работе нейронной сети (табл. 1, 2). Для исследования выбирают сети, показатели качества работы которых в номинальном режиме находятся в пределах заданных допусков.

3. На математических моделях имитируются вариации параметров элементов нейронов, заключающиеся в изменении значений весовых коэффициентов и пороговых смещений слоев на ±1 %. Процесс анализа включает рассмотрение последствий влияния данных изменений на качество (точность) работы нейронной сети. По формуле (1) рассчитывают значения K_i при использовании критерия максимальной относительной погрешности $E_{отн}$.

4. По выражению (3) рассчитывают значения $K_{ср}$ для каждого параметра и нейронной сети в целом, которые заносят в табл. 3, 4 и 5, 6 соответственно.

В соответствии с результатами, приведенными в табл. 1, в пределах заданных допусков функцию $y = \sin(\beta)$ аппроксимируют сети, обученные с помощью функции TRAINLM и TRAINBR. Изменяя значения весовых коэффициентов и пороговых смещений первого и второго слоев нейронных сетей на ±1 % вычислим значения критерия качества работы нейронной сети $K_{ср}$ для каждого параметра нейрона (табл. 3) и сети в целом (табл. 4).

Таблица 1

Показатели качества работы нейронных сетей в номинальном режиме, аппроксимирующих функцию $y = \sin(\beta)$

Функция обучения	Показатели качества работы НС при			
	6 нейронах в первом слое		8 нейронах в первом слое	
	$E_{абс}$	$E_{отн}$, %	$E_{абс}$	$E_{отн}$, %
TRAINLM	$1{,}17 \cdot 10^{-6}$	0,739	$2{,}62 \cdot 10^{-7}$	0,172
TRAINBR	$9{,}05 \cdot 10^{-8}$	0,090	$1{,}10 \cdot 10^{-7}$	0,085
TRAINSCG	$6{,}02 \cdot 10^{-3}$	6016,5	$9{,}45 \cdot 10^{-3}$	9454
TRAINGD	$5{,}57 \cdot 10^{-2}$	29053	$1{,}22 \cdot 10^{-1}$	14842
TRAINR	$3{,}38 \cdot 10^{-2}$	25640	$2{,}31 \cdot 10^{-2}$	23097
TRAINRP	$7{,}09 \cdot 10^{-3}$	7094,3	$7{,}11 \cdot 10^{-3}$	7110,2

Таблица 2

Показатели качества работы нейронных сетей, аппроксимирующих функцию $y = 1/\sqrt{x}$

Функция обучения	Показатели качества работы НС при			
	6 нейронах в первом слое		8 нейронах в первом слое	
	$E_{абс}$	$E_{отн}$, %	$E_{абс}$	$E_{отн}$, %
TRAINLM	$4{,}56 \cdot 10^{-6}$	0,000456	$1{,}77 \cdot 10^{-6}$	0,0001880
TRAINBR	$5{,}90 \cdot 10^{-8}$	0,00000625	$8{,}22 \cdot 10^{-7}$	0,0000912
TRAINSCG	$7{,}40 \cdot 10^{-3}$	0,7404	$5{,}38 \cdot 10^{-3}$	0,538
TRAINGD	$1{,}04 \cdot 10^{-1}$	10,355	$1{,}30 \cdot 10^{-1}$	18,346
TRAINR	$9{,}96 \cdot 10^{-3}$	0,996	$5{,}76 \cdot 10^{-3}$	0,624
TRAINRP	$2{,}80 \cdot 10^{-2}$	3	$8{,}44 \cdot 10^{-3}$	0,97

Таблица 3

Средний комплексный показатель качества $K_{ср}$ при изменении параметров нейронных сетей, аппроксимирующих функцию $y = \sin(\beta)$

Изменяемый параметр		TRAINLM		TRAINBR	
		Число нейронов в первом слое			
		6	8	6	8
Весовые коэффициенты первого слоя	−1 % +1 %	0,99 0,87	0,95 0,92	0,87 0,88	0,92 0,91
Весовые коэффициенты второго слоя	−1 % +1 %	−8388,77 −8388,72	−483,99 −483,91	−1335,64 −1335,67	−595,03 −594,99
Пороговые смещения первого слоя	−1 % +1 %	−507,49 −503,14	−158,44 −156,31	−101,44 −100,13	−199,65 −198,44
Пороговое смещение второго слоя	−1 % +1 %	−33768 −33766	−3319,4 −3319,6	−6467 −6467,1	−4072,9 −4072,9

Таблица 4

Средний комплексный показатель качества $K_{ср}$ работы нейронных сетей, аппроксимирующих функцию $y = \sin(\beta)$

Функция обучения		Число нейронов в первом слое	
		6	8
TRAINLM	−1 % +1 %	−4586,28 −4584,84	−338,05 −337,36
TRAINBR	−1 % +1 %	−793,91 −793,51	−416,92 −416,52

В соответствии с результатами, приведенными в табл. 2, в пределах заданных допусков функцию $y = 1/\sqrt{x}$ аппроксимируют сети, обученные с помощью функции TRAINLM, TRAINBR, TRAINSCG и TRAINR. Изменяя значения весовых коэффициентов и пороговых смещений первого и второго слоев нейронных сетей на ±1 % вычислим критерий качества работы нейронной сети $K_{ср}$ для каждого параметра нейрона (табл. 5) и сети в целом (табл. 6).

Таблица 5

Средний комплексный показатель качества $K_{ср}$ при изменении параметров нейронных сетей, аппроксимирующих функцию $y = 1/\sqrt{x}$

Изменяемый параметр		TRAINLM		TRAINBR	
		Число нейронов в первом слое			
		6	8	6	8
Весовые коэффициенты первого слоя	−1 %	0,94	0,95	0,93	0,96
	+1 %	0,95	0,95	0,93	0,96
Весовые коэффициенты второго слоя	−1 %	−1,01	0,98	0,56	0,95
	+1 %	−1,01	0,98	0,56	0,95
Пороговые смещения первого слоя	−1 %	0,95	0,98	0,96	0,94
	+1 %	0,95	0,98	0,96	0,94
Пороговое смещение второго слоя	−1 %	−4,52	0,55	−2,10	0,53
	+1 %	−4,52	0,55	−2,10	0,53
		TRAINSCG		TRAINR	
Весовые коэффициенты первого слоя	−1 %	0,94	0,86	0,70	0,74
	+1 %	0,94	0,87	0,77	0,71
Весовые коэффициенты второго слоя	−1 %	0,99	0,99	1,00	0,99
	+1 %	1,00	1,00	0,99	1,00
Пороговые смещения первого слоя	−1 %	0,95	0,87	0,78	0,71
	+1 %	0,94	0,85	0,70	0,73
Пороговое смещение второго слоя	−1 %	0,28	0,43	0,05	0,31
	+1 %	0,13	0,39	0,03	0,31

Таблица 6

Средний комплексный показатель качества $K_{ср}$ работы нейронных сетей, аппроксимирующих функцию $y = 1/\sqrt{x}$

Функция обучения		Число нейронов в первом слое	
		6	8
TRAINLM	−1 %	0,04	0,95
	+1 %	0,04	0,95
TRAINBR	−1 %	0,66	0,93
	+1 %	0,66	0,93
TRAINSCG	−1 %	0,92	0,89
	+1 %	0,92	0,89
TRAINR	−1 %	0,79	0,79
	+1 %	0,78	0,79

Анализ полученных результатов позволяет сделать вывод, что нейронные сети, аппроксимирующие функцию $y = \sin(\beta)$, чувствительны к вариациям значений параметров нейронов второго слоя, а также к изменению пороговых смещений нейронов первого слоя (см. табл. 3). При этом, хотя общее качество работы нейронной сети и улучшается с увеличением числа нейронов в первом слое (см. табл. 4), исследуемые нейронные сети не удовлетворяют техническим требованиям, так как качество их работы при вариациях значений параметров, входящих в них элементов выходит за границы поля допуска. Нейронные сети, аппроксимирующие функцию $y = 1/\sqrt{x}$, с 8 нейронами в первом слое сохраняют свою работоспособность в пределах заданного допуска (табл. 6). При снижении числа нейронов в первом слое до 6 качество работы нейронных сетей, обученных по TRAINLM и TRAINBR, выходит за границы поля допуска при изменении значений пороговых смещений нейронов второго слоя НС (см. табл. 5). В соответствии с результатами, приведенными в табл. 6, исследуемые нейронные сети удовлетворяют техническим требованиям, так как качество (точность) их работы при вариациях значений параметров их элементов не выходит за границы поля допуска.

Таким образом, критерий качества K позволил определить наилучшие функции обучения для нейронных сетей, решающих поставленные технические задачи.

Результаты проведенных исследований позволяют сделать вывод, что комплексный показатель K, учитывающий значения функциональных допусков, является инвариантным к структуре и сложности нейронных сетей, а также к характеру решаемых ими задач. Он позволяет эффективно оценивать уровень качества (точности) работы нейронных сетей различного назначения.

Работа выполнена при поддержке гранта РФФИ № 11-08-97551.

Список литературы

1. **Галушкин А. И.** Теория нейронных сетей. М.: ИПРЖР, 2000. 416 с.

2. **Медведев В. С., Потемкин В. Г.** Нейронные сети. MATLAB 6/ Под общ. ред. Потемкина В. Г. М.: ДИАЛОГ-МИФИ, 2002. 496 с.

3. **Димов Ю. В.** Метрология, стандартизация и сертификация: учебник. СПб.: Питер, 2005. 432 с.

4. **Данилин С. Н., Макаров М. В., Щаников С. А.** Исследование коэффициентов влияния погрешностей элементов нейронов на показатели точности (качества) работы устройств с нейросетевой архитектурой // Алгоритмы, методы и системы обработки данных: электронный научный журнал. 2011. URL: http://www.amisod.ru/index.php.

5. **Данилин С. Н., Макаров М. В., Щаников С. А.** Алгоритм определения коэффициентов влияния погрешностей элементов нейронов на показатели качества работы устройств с нейросетевой архитектурой // Методы и устройства передачи и обработки информации. 2011. Вып. 13, № 1. Муром: МИВлГУ. С. 114—118.

УДК 519.226.3:336.201.2

С. А. Горбатков, д-р техн. наук, проф.,
О. Б. Рашитова, ст. преподаватель,
e-mail: 286623@rambler.ru,
Финансовый университет при правительстве РФ

Моделирование налоговых управленческих решений на основе нейронных сетей Кохонена

Рассматривается подход к повышению эффективности принятия решений налогового регулирования на основе нейросетевого моделирования (самоорганизующихся карт Кохонена). Предложен метод селекции факторов при оценке финансового состояния предприятий — налогоплательщиков в процедуре их нейросетевой кластеризации на основе байесовского подхода. Представлены результаты исследований эффективности предложенного метода на примере оценки кредитоспособности предприятий сельского хозяйства.

Ключевые слова: *нейронные сети, налоговое администрирование, кластеризация, селекция факторов, байесовский подход*

Введение

В практике работы налоговых органов местного уровня зачастую возникает необходимость оперативного принятия решений по налоговому регулированию для большого числа предприятий — налогоплательщиков (порядка сотен и тысяч). При этом времени на подробный, углубленный анализ каждого предприятия нет. В этом случае предлагается проводить оперативную групповую (кластерную) оценку финансового состояния налогоплательщиков с помощью нейросетей Кохонена. Например, решения по налоговому регулированию разрабатываются в пространстве пяти кластеров, где соответственно группируются налогоплательщики с "очень хорошим", "хорошим", "средним", "плохим" и "очень плохим" уровнем кредитоспособности. Последний оценивается по совокупности показателей (признаков), которые могут быть легко определены из бухгалтерской отчетности налогоплательщиков.

Таким образом, мы приходим к задаче кластеризации предприятий — налогоплательщиков в пространстве множества признаков x_j, $j = \overline{1, n}$. Векторы $\mathbf{x}_i = (x_{1i}, x_{2i}, ..., x_{ni})$, где $i = 1, 2, ..., N$ — номер наблюдения, будем далее называть векторами факторов.

Общая теория задач кластеризации хорошо изучена и базируется в основном на предположении о соответствии исходных данных (векторов $\{\mathbf{x}_i\}$) гауссовым смесям распределения плотности вероятности кластеризируемых объектов [1]. Однако на практике это предположение зачастую не выполняется. При использовании нейросетевых инструментариев, например, самоорганизующихся карт Кохонена (SOM-карт) [2] появляется еще одна проблема — обеспечение устойчивости нейросетевой модели при вариации параметров обучения сети. Применительно к задачам налогового регулирования эта проблема весьма актуальна, поскольку обучение нейросети проводится на сильно зашумленной (вплоть до сознательного искажения) базе данных $\{\mathbf{x}_i\}$. Регуляризация нейросетей Кохонена в предположении о том, что $\{\mathbf{x}_i\}$ представляют собой гауссовы смеси, подробно рассматривалась в работе [3]. Отметим, что для практического применения допущение о гауссовском распределении указанных смесей является весьма "жестким".

Помимо регуляризации моделей дополнительная прикладная проблема в рассматриваемом классе задач нейросетевой кластеризации — это необходимость сокращения размерности n признакового пространства [4, 5], т. е. выделения из большого числа факторов наиболее существенных.

В данной работе предлагается сократить размерность пространства признаков и повысить адекватность используемых моделей за счет селекции используемых признаков кластеризации на фоне регуляризации нейронных сетей на основе оригинального квазибайесовского метода (КБМ).

Кластеризация налогооблагаемых предприятий в целях принятия решений по налоговому регулированию повышает объективность принимаемых решений. Лицо, принимающее решение, может отнести то или иное предприятие к одному из получаемых кластеров и принять правильное решение по результатам такого математического моделирования.

Селекция признаков кластеризации

В вышеуказанных прикладных задачах кластеризации вектор признаков $\mathbf{x} = (x_1, x_2, ..., x_n)$ может содержать большое число компонент (до нескольких десятков и сотен). Часть из этих компонент имеют высокую информативность в аспекте разделения векторов (объектов) $\mathbf{x}$ на классы, а часть малоинформативна. В результате качество анализа с использованием моделей систем искусственного интеллекта, например кластеризации, как в нашем случае, может оказаться неудовлетворительным.

В связи с этим предлагается метод кластеризации с селекцией признаков и байесовской регуляризацией (КСП и БР), предусматривающий итерационный процесс кластеризации с текущей оценкой ее качества [5].

Селекция признаков при кластеризации представляет собой процедуру выделения из всего мно-

жества признаков меньшего подмножества с сохранением информативности. Суть селекции признаков — это выделение признаков, которые приводят к большим расстояниям между классами и к малым расстояниям внутри классов, т. е. к минимизации критерия качества

$$\Theta_q = \frac{\left(\sum_{m=1}^{M} \sum_{i=1}^{N_m} d^2(\mathbf{x}_{im}, \mathbf{x}_{\text{ц}m}) \right)}{\left(\frac{1}{C_M^2} \sqrt{\sum_{l=1}^{M} \sum_{m=l+1}^{M} d^2(\mathbf{x}_{\text{ц}l}, \mathbf{x}_{\text{ц}m})} \right)}, \qquad (1)$$

где $q = 1, 2, ..., Q$ и Q — соответственно номер гипотезы-нейросети в байесовском ансамбле и их общее число; N_m — число элементов, попавших в m-й кластер; $\mathbf{x}_{\text{ц}m}$ — точка центра m-го кластера в n-мерном евклидовом пространстве признаков; $d(\mathbf{x}_{im}; \mathbf{x}_{\text{ц}m})$ — евклидово расстояние от исследуемого объекта $\mathbf{x}_{im}$ до центра своего m-го кластера; $d(\mathbf{x}_{\text{ц}l}; \mathbf{x}_{\text{ц}m})$ — расстояние между l-м и m-м кластерами; C_M^2 — число сочетаний из M по 2; M — число кластеров.

В предлагаемом методе используется скалярная селекция признаков, предусматривающая отдельное независимое рассмотрение используемых признаков.

Суть метода скалярной селекции признаков состоит в оценке дискриминантной способности каждого отдельного признака x_j путем проверки соответствующих статистических гипотез о законах распределения плотности вероятности анализируемого признака в разных классах (кластерах). Если распределения плотности $f(x_j|\Omega_l)$, $l = 1, \overline{M}$, совпадают для разных классов ($l \neq m$) при назначенной мере сходства, то признак не различает эти классы Ω_l и Ω_m.

Рассмотрим количественные соотношения для алгоритма скалярной селекции для наглядности на примере двух классов: Ω_1 и Ω_2, хотя все соотношения, приводимые ниже, справедливы для любого конечного числа классов $\Omega_1, \Omega_2, ..., \Omega_m, ..., \Omega_M$.

Пусть известны прецеденты для признака x_j, т. е. случайные его реализации из класса Ω_1 $\{x_{ij,1}\} \in \Omega_1$; $i = \overline{1, N_1}$, и для класса Ω_2 $\{x_{ij,2}\} \in \Omega_2$; $i = \overline{1, N_2}$, где N_1, N_2 — числа прецедентов в классах Ω_1 и Ω_2. Методами математической статистики [6] оцениваются эмпирические законы распределения плотности вероятности для $\{x_{ij,1}\}$ и $\{x_{ij,2}\}$.

Обозначим через H_0 и H_1 две гипотезы: H_1 — средние значения признаков различаются существенно в классах Ω_1 и Ω_2; H_0 — средние значения признаков различаются несущественно — нуль-гипотеза.

Примем соглашение о том, что признаки $\{x_j\}$ при селекции нормализованы стандартным способом [7] и пусть известны дисперсии нормированных признаков $\tilde{x}_j$ для прецедентов в классах. Тогда при селекции основная задача — проверить отличие средних значений μ_1 и μ_2 признаков в двух классах Ω_1 и Ω_2. Соответствующие гипотезы имеют вид: H_0: $\Delta\mu_j = \mu_{j1} - \mu_{j2} = 0$; H_1 : $\Delta\mu_j \neq 0$.

В задаче о значимости различия средних μ_{j1} и μ_{j2} используется критерий Стьюдента [6] с числом степеней свободы $(N_1 + N_2) - 2$:

$$t_j = \frac{|\mu_{j1} - \mu_{j2}|}{S_j \sqrt{\frac{1}{N_1} + \frac{1}{N_2}}}, \qquad (2)$$

где S_j — общее среднеквадратическое отклонение анализируемого признака от своих средних в кластерах Ω_1 и Ω_2:

$$S_j = \frac{(N_1 - 1)S_{j1}^2 + (N_2 - 1)S_{j2}^2}{(N_1 + N_2) - 2}.$$

Заметим, что если закон распределения плотности вероятности анализируемого признака x_j в кластерах Ω_1 и Ω_2 отклоняется от нормального, то оценка дискриминантной способности этого признака будет носить приближенный характер. Для уменьшения погрешности такой оценки в предлагаемом методе используется байесовский подход к регуляризации модели кластеризации, подробно описанный ниже.

После вычисления критерия Стьюдента по формуле (2) проверяем нуль-гипотезу:

$$H_0 : t_j \leqslant t_\alpha(\alpha; \nu = N_1 + N_2 - 2)? \qquad (3)$$

Здесь t_α — табличное значение критерия Стьюдента с принятым уровнем значимости α; ν — число степеней свободы для статистики Стьюдента.

Если неравенство (3) выполнено, то делается вывод о том, что статистически значимого различия в средних μ_1 и μ_2 по классам Ω_1 и Ω_2 нет. Следовательно, анализируемый признак x_j неинформативен в аспекте разделения данных на классы (кластеры) с принятым уровнем значимости α и при имеющихся объемах прецедентов в двух классах N_1 и N_2. При конечном числе классов M описанная процедура селекции признаков проводится последовательно попарно для классов Ω_l, Ω_m, $(m \neq l)$.

Признак $\{x_j\}$ считается неинформативным для выполненного разбиения, если он не различает распределения этого признака более чем в одном из всех сочетаний пар классов Ω_l, Ω_m, $(m \neq l)$; $l, m = 1, 2, ..., M$.

Байесовский подход к регуляризации нейронной сети

Причиной неудовлетворительного качества кластеризации с помощью нейросетевых инструментариев является возможная сильная зависимость результатов кластеризации от параметров настройки

сети. Так, для рассматриваемого примера использования в качестве нейросетевого кластеризатора самонастраивающейся карты Кохонена (SOM) результаты кластеризации оказались сильно зависящими от параметров, определяющих динамику скорости ее обучения, и величины гауссовой окрестности возбужденного нейрона. Данная неустойчивость нейросетевого кластеризатора по вариации параметров настройки в методе КСП и БР парируется на основе байесовской процедуры регуляризации нейросети.

Главные идеи байесовского подхода [3], используемые в предлагаемом методе следующие:

- выбор ансамбля априорных гипотез-нейросетей $\{h_q(\mathbf{x}, W)\}$, где W — множество параметров модели, осуществляется из фиксированного класса (семейства) H метагипотез (сетей Кохонена);
- апостериорная фильтрация обученных гипотез-нейросетей осуществляется по критерию, оценивающему качество кластеризации (1) как по плотности группировки объектов вокруг центров кластеров (числитель отношения (1)), так и по удаленности кластеров друг от друга (знаменатель в (1));
- после фильтрации гипотез-нейросетей осуществляется усреднение критерия качества разбиения векторов (объектов) $\mathbf{x} \in D$ на кластеры по формуле (1) на отфильтрованном ансамбле гипотез-нейросетей.

В предлагаемом методе КСП и БР разработки нейронной сети формула Байеса непосредственно не используется для апостериорной оценки вероятности $\{P(h_q|D|H)\}$ выбранных гипотез-нейросетей, где $D = \{x_{ij}\}_{i=1}^{i=N}$, ($j = \overline{1, n}$) — совокупность вектор-строк данных, поскольку для оценки указанной вероятности через функцию правдоподобия требуется априорное знание аналитической формы закона распределения кластеризуемых векторов $\mathbf{x}$, например в виде гауссовой смеси. Такого знания у нас нет. Поэтому апостериорные вероятности $\{P(h_q|D|H)\}$, несущие информацию о качестве разбиения данных D на кластеры, в предлагаемом методе КСП и БР оцениваются косвенно путем фильтрации гипотез-нейросетей $\{h_q\}$ по критерию (1).

При фильтрации гипотез-нейросетей организуется итерационный процесс пошагового отбора (удаления из ансамбля) гипотез-нейросетей $\{h_q\}$ с низким качеством кластеризации (1), т. е. большим значением $\Theta_q[h_q(\mathbf{x}, W)|D|H]$:

$$q^*: \Theta^{(q)} \leqslant \Theta_0;\ q = 1, 2, ..., Q, \qquad (4)$$

где q^* — номер гипотезы-нейросети, успешно прошедшей процедуру фильтрации; Θ_0 — желаемое значение качества фильтрации, определяемое в предварительных вычислительных экспериментах.

После фильтрации (4) уточненные значения центров кластеров $\{\mathbf{x}_{\text{ц}m}\}$ и соответствующего им критерия качества разбиения Θ по (1) находятся как усредненные на отфильтрованном байесовском ансамбле величины:

$$\mathbf{x}_{\text{ц}m} = \left[\sum_{q^*=1}^{Q^*} \mathbf{x}_{\text{ц}mq^*}\right] / Q^*;\ \overline{\Theta}^* = \left(\sum_{q^*=1}^{Q^*} \Theta_{q^*}\right) / \Theta^*. \qquad (5)$$

Процесс обучения SOM характеризуется, во-первых, окрестностью взаимодействия k^*-го нейрона с i-м вектором обучающей выборки [2]:

$$h_{k^*i} = \exp\left(-\frac{d_{k^*i}^2}{2\sigma^2}\right),$$

где d_{k^*i} — расстояние взаимодействия по евклидовой метрике; σ — параметр гауссового распределения $\sigma(t) = \sigma_0 \exp\left(-\frac{t}{\tau_1}\right)$, $t = 0, 1, 2, ...$, σ_0 — начальное значение величины σ в алгоритме SOM; τ_1 — некоторый параметр, влияющий на характеристику σ. Во-вторых, процесс обучения SOM определяется скоростью изменения (модификации) весов при обучении, характеризуемой параметром $\eta(t)$, экспоненциально изменяющегося от номера повторного прогона обучающей выборки (фактически от времени t):

$$\eta(t) = \eta_0 \exp\left(-\frac{t}{\tau_2}\right),\ t = 0, 1, 2, ...,$$

где τ_2 — еще один параметр, влияющий на эффективность работы алгоритма SOM.

Процесс фильтрации описывается следующим алгоритмом.

Шаг 0. $k = 0$ (k — номер итерации), $n^{(k)} = n$. При заданных M, Θ_0 и полной размерности n вектора признаков $\mathbf{x} \in R^n$ осуществляем начальное разбиение D на кластеры методом самоорганизующихся карт Кохонена [2]. При этом строится байесовский ансамбль из Q нейросетей Кохонена $\{h_q(\mathbf{x}, W)|D|H\}$ и по формулам (3)—(4) находятся уточненные центры кластеров $\mathbf{x}_{\text{ц}m}^{(k)}$ и критерий качества разбиения $\overline{\Theta}^{*(k)}$.

Шаг 1. $k = k + 1$. Проводим по каждому признаку x_j расчет критериев Стьюдента t_j по формуле (2) и проверяем соответствующие нуль-гипотезы H_0 по формуле (3) для всех пар образованных кластеров Ω_l, Ω_m, $(m \neq l)$; $l, m = 1, 2, ..., M$.

Шаг 2. Проводим проверку информативности каждого признака исходя из условия, что он различает распределения этого признака более чем в одном из всех сочетаний пар классов Ω_l, Ω_m, $(m \neq l)$; $l, m = 1, 2, ..., M$.

Шаг 3. Если все рассмотренные признаки информативны, то процедура селекции заканчивает-

ся, если нет — осуществляется исключение неинформативных признаков из компонент вектора **x**, т. е. сокращение размерности вектора **x** до $n^{(k)} = n^{(k-1)} - u$, где u — число исключенных на данной итерации признаков.

Шаг 4. Осуществляем кластеризацию с измененным составом признаков и проверяем условие улучшения качества разбиения на k-й итерации селекции признаков:

$$\overline{\Theta}^{*(k)} < \overline{\Theta}^{*(k-1)}? \qquad (6)$$

Если неравенство (6) выполняется, то, продолжая процесс отбора, переходим к шагу 1, если нет — "ужесточаем" правила отбора по формулам (3) и (4) и, начиная заново процесс отбора, переходим к шагу 0.

Замечание. Как известно [2], результат кластеризации с помощью сети Кохонена изменяется при повторных запусках нейросети (дублирующих расчетах) за счет случайного характера выбора начальных положений нейронов. В этом случае в обобщенном критерии качества разбиения (5) полезно вместо $\{\Theta_{q^*}\}$ брать значения, осредненные по дублирующим расчетам: $\Theta_{q^*} \to \overline{\Theta}_{q^*} = \frac{1}{R}\sum_{t=1}^{R}\Theta_{q^*,r}$, где $\Theta_{q^*,r}$ — значение критерия качества в r-м повторном расчете; R — число повторов.

Практические результаты

Апробация предлагаемого метода КСП и БР проводилась на реальных (закодированных) данных сельскохозяйственных предприятий-налогоплательщиков Республики Башкортостан [4]. В качестве признаков кластеризации использованы показатели, достаточно полно характеризующие состояние предприятий в условиях российской экономики. Они отражают мнения руководителей коммерческих предприятий (по результатам их опроса) и входят в состав методики Федерального управления по делам несостоятельности (банкротства) и моделей Альтмана [8].

В качестве основных используются 16 показателей. *Первую группу* составляют показатели, характеризующие рентабельность предприятия:

- R1 — общая рентабельность (отношение балансовой прибыли к сумме выручки от продаж и внереализационных доходов);
- R2 — рентабельность активов (отношение чистой прибыли к средней балансовой стоимости активов);
- R3 — рентабельность собственного капитала (отношение чистой прибыли к сумме доходов будущих периодов, капиталов и резервов (за вычетом собственных акций, выкупленных у акционеров) за вычетом целевого финансирования и поступлений);
- R4 — рентабельность продукции (отношение прибыли от продаж к выручке от продаж);
- R5 — рентабельность оборотных активов (отношение чистой прибыли к средней стоимости оборотных активов).

Вторую группу составляют показатели, характеризующие ликвидность и платежеспособность предприятия:

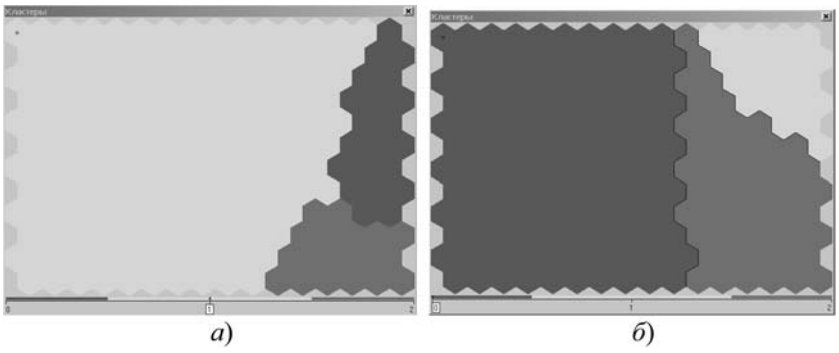
- L1 — быстрый коэффициент ликвидности (отношение вычета запасов, налога на добавленную стоимость по приобретенным ценностям и долгосрочной дебиторской задолженности из оборотных активов к краткосрочным обязательствам, не включая доходы будущих периодов);
- L2 — коэффициент покрытия запасов (отношение суммы оборотных собственных средств, краткосрочных займов, кредитов и краткосрочной кредиторской задолженности к средней величине запасов);
- L3 — текущий коэффициент ликвидности (отношение разности оборотных активов и долгосрочной дебиторской задолженности к краткосрочным обязательствам, не включая доходы будущих периодов).

Третью группу составляют показатели, характеризующие деловую активность:

- A2 — оборачиваемость активов (отношение выручки от продажи за вычетом налога на добавленную стоимость, акцизов и других обязательств к средней стоимости активов);
- A4 — оборачиваемость кредиторской задолженности (отношение выручки от продажи без учета коммерческих и управленческих расходов к средней кредиторской задолженности);
- A5 — оборачиваемость дебиторской задолженности (отношение выручки от продажи за вычетом налога на добавленную стоимость, акцизов и других обязательств к разности дебиторской задолженности на конец отчетного периода и задолженности учредителей по вкладам в уставной капитал на конец отчетного периода);
- A6 — оборачиваемость запасов (отношение себестоимости к средней величине запасов).

Четвертую группу составляют показатели, характеризующие финансовую устойчивость предприятия:

- F1 — коэффициент финансовой зависимости (отношение суммы долгосрочных и краткосрочных обязательств, не включая доходы будущих периодов к сумме доходов будущих периодов, капитала и резервов (за вычетом собственных акций, выкупленных у акционеров) за вычетом целевого финансирования и поступлений);
- F2 — коэффициент автономии собственных средств (отношение суммы доходов будущих периодов, капитала и резервов (за вычетом собственных акций, выкупленных у акционеров) за



а) *б*)

Рис. 1. Результаты кластеризации по всей совокупности признаков (*а*) и после селекции признаков (*б*)

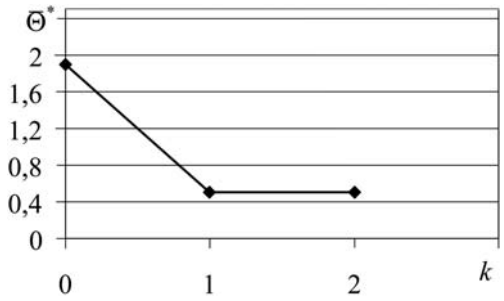


Рис. 2. Динамика критерия качества в процессе селекции

вычетом целевого финансирования и поступлений к сумме внеоборотных и оборотных активов);

- F3 — обеспеченность запасов собственными оборотными средствами (отношение собственных оборотных средств к запасам);
- F4 — индекс постоянного актива (отношение суммы внеоборотных активов и долгосрочной дебиторской задолженности к сумме доходов будущих периодов, капитала и резервов (за вычетом собственных акций, выкупленных у акционеров) за вычетом целевого финансирования и поступлений).

Практическое подтверждение предложенного метода селекции признаков осуществлялось для группы из 24 предприятий, для которых были просчитаны вышеназванные 16 показателей в качестве признаков кластеризации. В качестве инструментария кластеризации использовалась нейронная сеть Кохонена, которая успешно применяется в задаче принятия решений при налоговом администрировании [4, 9, 10].

В рассматриваемом примере в гипотезах-нейросетях $\{h_q(\mathbf{x}, W)|\mathrm{H}\}$ варьировались две вышеназванные эвристики — τ_1 и τ_2. Параметр $\{\tau_1\}$ варьировался дискретно на трех уровнях: $\tau_1 = \{140; 280; 700\}$. Параметр $\{\tau_2\}$ варьировался дискретно также на трех уровнях: $\tau_2 = \{125; 250; 625\}$.

Уровни указанных параметров подбирались путем предварительных вычислительных экспериментов. В различных сочетаниях уровней τ_1 и τ_2 было образовано в байесовском ансамбле девять сетей Кохонена. Для каждой q-й сети проводилось по три дублирующих обучения.

Для расчетов использовался программный продукт Deductor Studio 4.4 (демоверсия с ограничением числа записей) для аналитической платформы Deductor Lite.

Проводились предварительные вычислительные эксперименты по выбору параметров адаптивного процесса обучения сети Кохонена в целях определения начального значения ширины σ_0 функции топологической окрестности $h_{k^*,q}(d_{k^*,q})$, начальной скорости обучения η_0, числа эпох T (итераций) процесса модификации весов: $T = 500$; $\sigma_0 = 4$; $\eta_0 = 0{,}3$.

Основные результаты процесса кластеризации в представленном ансамбле гипотез-нейросетей заключаются в следующем.

Фильтрация байесовских гипотез $\{h_q(\mathbf{x}, W)|H\}$ для байесовского метода кластеризации 24 рассмотренных предприятий проведена во всех 27 нейросетевых моделях байесовского ансамбля.

Обобщенный показатель Θ по формуле (1), оценивающий косвенно апостериорную вероятность гипотез-нейросетей $P(h_q(\mathbf{x}, W)|\mathrm{D}|\mathrm{H})$, изменяется на множестве из 27 сетей ансамбля в довольно широких пределах: от $\Theta_q[(h_q)]_{\min} = 0{,}43852$ до $\Theta_q[(h_q)]_{\max} = 0{,}59826$, т. е. на 36,4 %.

Согласно байесовскому методу было использовано предложенное правило фильтрации (4).

Уже на первой итерации фильтрации установлено, что все девять сравниваемых гипотез-нейросетей $\{h_q\}$ прошли условия отбора и оставлены в апостериорном (отфильтрованном) ансамбле.

После выполнения первой итерации процедуры селекции (рис. 1, *а*) при кластеризации с использованием всех признаков были оставлены в качестве информативных только пять: L2, A6, F1, F2, F3, т. е. размерность n пространства признаков уменьшена в 3,2 раза.

На второй итерации (рис. 1, *б*) оценка информативности этих признаков не изменилась с сохранением качества кластеризации по формуле (1).

Таким образом, было сокращено пространство признаков кластеризации для их практического применения. Усредненное по формуле (5) значение критерия качества значительно уменьшилось (рис. 2), что говорит об эффективности процедуры селекции, а окончательно полученное значение $\overline{\Theta}^{*(2)} = 0{,}505$ свидетельствует о достаточно высоком качестве разбиения.

Заключение

Показана эффективность предложенного метода селекции признаков кластеризации, выраженная в значительном сокращении их числа, в рассматриваемом примере моделирования принятия решений при налоговом администрировании предприятий.

Применительно к условиям моделирования рассмотренного примера (сильное зашумление данных и их дефицит) проявляется заметное расслоение результатов разбиения на кластеры в зависимости от параметров адаптивного обучения сети Кохонена, которые в широких пределах варьировались в ансамбле сетей. Следовательно, идея о необходимости регуляризации нейросетевого классификатора подтвердилась в вычислительных экспериментах. Идея байесовского метода регуляризации количественно апробирована и подтверждена для рассмотренной практической задачи.

Направление дальнейших исследований авторы видят в изучении отраслевой специфики предложенного метода.

Список литературы

1. **Патрик Э. А.** Основы теории распознавания образов / Пер. с англ.; под ред. Б. Р. Левина. М.: Сов. радио, 1980. 408 с.
2. **Хайкин С.** Нейронные сети: полный курс / Пер. с англ.; 2-е изд. М.: Издательский дом "Вильямс", 2006. 1104 с.
3. **Шумский С. А.** Байесова регуляризация обучения // Лекции по нейроинформатике. Часть 2. М.: МИФИ, 2002. С. 30—93.
4. **Нейросетевое** математическое моделирование в задачах ранжирования и кластеризации в бюджетно-налоговой системе регионального и муниципального уровней / С. А. Горбатков, Д. В. Полупанов, А. М. Солнцев, И. И. Белолипцев, М. В. Коротнева, С. А. Фархиева, О. Б. Рашитова. Уфа: РИД БашГУ, 2011. 222 с.
5. **Св. ОФЭРНИО 17538.** Байесовский итерационный алгоритм кластеризации на основе селекции признаков / С. А. Горбатков, О. Б. Рашитова. Рег. 31.10.2011.
6. **Айвазян С. А., Мхитарян В. С.** Прикладная статистика и основы эконометрики. М.: ЮНИТИ, 1998. 1023 с.
7. **Ежов А. А., Шумский С. А.** Нейрокомпьютинг и его применение в экономике и бизнесе: Учебник / Под ред. проф. В. В. Харитонова. М.: Изд-во МИФИ, 1998. 224 с.
8. **Давыдова Г. В., Беликов А. Ю.** Методика количественной оценки риска банкротства предприятий // Управление риском. 1999. № 3. С. 13—20.
9. **Горбатков С. А., Рашитова О. Б.** Кластеризация заемщиков для оценки кредитного риска // Проведение научных исследований в области обработки, хранения, передачи и защиты информации: Сб. науч. тр. в 4 т. Т. 4. Ульяновск: УлГТУ, 2009. С. 394—403.
10. **Горбатков С. А., Полупанов Д. В., Макеева Е. Ю., Бирюков А. Н.** Методологические основы разработки нейросетевых моделей экономических объектов в условиях неопределенности / Под ред. проф. С. А. Горбаткова. М.: Издательский дом "Экономическая газета", 2012. 490 с.

УДК 004.932:681.5

Д. А. Юдин, аспирант,
e-mail: yuddim@yandex.ru,
В. З. Магергут, д-р техн. наук, проф.,
Белгородский государственный
технологический университет им. В. Г. Шухова

Сегментация изображений процесса обжига с применением текстурного анализа на основе самоорганизующихся карт

Разработан способ сегментации изображений процесса обжига с применением текстурного анализа на основе самоорганизующихся карт. Осуществлена программная реализация предлагаемого способа. Приведены функциональные возможности разработанного программного комплекса. Найден набор текстурных характеристик, обеспечивающих квазиоптимальную сегментацию изображения процесса обжига. Приведено сравнение сегментации изображения на основе самоорганизующихся карт с методом k-средних. Показаны преимущества применения аппарата самоорганизующихся карт для классификации векторов текстурных характеристик.

Ключевые слова: *сегментация изображений, процесс обжига, вращающиеся печи, самоорганизующиеся карты, нейронные сети, текстурный анализ, программный комплекс*

Введение

Анализ визуальной информации о процессе обжига сырья во вращающихся печах цементных и керамзитовых заводов позволяет снизить избыточный расход топлива и повысить качество конечного продукта (цементного клинкера или керамзитового гравия) [1]. Автоматизированный анализ задачи требует применения систем технического зрения, а соответственно, возникает необходимость обработки и распознавания изображений процесса обжига, в том числе для мониторинга и управления печами.

Из рис. 1, *а* видно, что на видеоизображении процессов практически нет естественных контуров, отделяющих материал от пламени, материал от футеровки печи и футеровки от корпуса печи.

Для решения задачи сегментации изображений целесообразно применить текстурный анализ с использованием самоорганизующихся карт [2]. Этот анализ позволяет осуществить сегментацию изображения, т. е. разделить изображение на области различных типов, например, "материал", "пламя", "футеровка", "корпус" (рис. 1, *б*) и др. Затем для каждого из сегментов могут быть вычислены геометрические, качественные и яркостные признаки, на основе которых формируются выходные образы, используемые операторами печи для выработки технологических рекомендаций и непосредственного мониторинга вращающейся печи (последнее в данной статье не рассматривается).

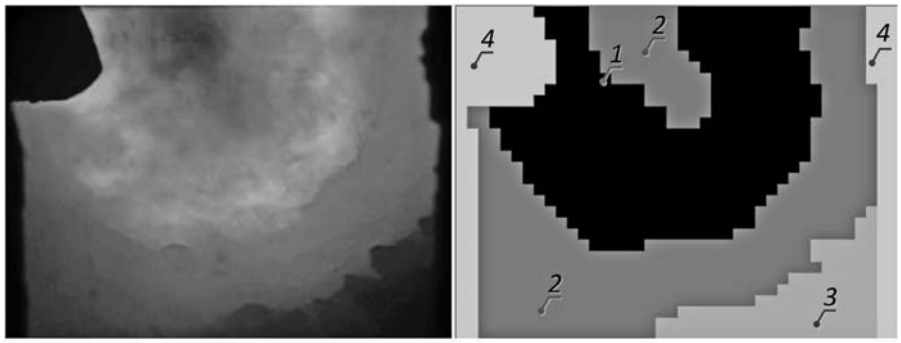


Рис. 1. Исходные данные для алгоритмов сегментации:
а — исходное изображение процесса обжига во вращающейся печи; *б* — эталон сегментации исходного видеоизображения на четыре класса (*1* — черный — сегмент факела; *2* — темно-серый — сегмент материала; *3* — серый — сегмент футеровки; *4* — светло-серый — сегмент корпуса печи)

Основной проблемой, возникающей при решении задачи анализа текстур, является определение набора характеристик достаточного для описания пространственной текстуры, присутствующей в изображении. Формальной процедуры задания исходного набора характеристик текстуры пока не существует. Характеристики, используемые при решении тех или иных задач, задаются лишь на основании опыта и интуиции специалиста. В рассматриваемом случае используется 11 текстурных характеристик, большинство из которых строится на основе матрицы смежности области изображения [3, 4].

1. Энергия: $\sum_{i=0}^{K-1}\sum_{j=0}^{K-1}(P_{ij}^{d})^2$,

где K — число градаций яркости изображения; P^d — матрица смежности; d — расстояние между точками при подсчете матрицы смежности.

2. Контраст: $\sum_{i=0}^{K-1}\sum_{j=0}^{K-1}|i-j|P_{ij}^{d}$.

3. Корреляция: $\sum_{i=0}^{K-1}\sum_{j=0}^{K-1}\frac{(i-\mu)(j-\mu)P_{ij}^{d}}{\sigma^2}$,

где μ и σ^2 — соответственно математическое ожидание и дисперсия элементов матрицы смежности P_{ij}^{d}.

4. Автокорреляция: $\sum_{i=0}^{K-1}\sum_{j=0}^{K-1}(ij)P_{ij}^{d}$.

5. Однородность: $\sum_{i=0}^{K-1}\sum_{j=0}^{K-1}\frac{1}{1+|i-j|}P_{ij}^{d}$.

6. Энтропия: $\sum_{i=0}^{K-1}\sum_{j=0}^{K-1}P_{ij}^{d}\ln(P_{ij}^{d})$.

7. Инерция: $\sum_{i=0}^{K-1}\sum_{j=0}^{K-1}(i-j)^2P_{ij}^{d}$.

8. Тень: $\sum_{i=0}^{K-1}\sum_{j=0}^{K-1}(i+j-2\mu)^3P_{ij}^{d}$.

9. Максимальная вероятность: $\max_{i,j}P_{ij}^{d}$.

10. Интенсивность: $\frac{1}{N^2}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}f(i,j)$,

где $f(i, j)$ — яркость пикселя в точке (i, j) скользящего окна размером $N \times N$.

11. Вариация: $\sum_{g=0}^{K-1}\left(g-\frac{1}{N^2}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}f(i,j)\right)C(g)$,

где $C(g)$ — число пикселей с яркостью g.

На основании этих текстурных характеристик необходимо сформировать набор, обеспечивающий эффективную сегментацию изображения процесса обжига.

Постановка задачи

В качестве входного изображения используется изображение процесса обжига (рис. 1, *а*) разрешением 800 × 600, которое разбивается на области размером 20 × 20 пикселей. Такое разбиение является приемлемым для задачи распознавания изображения процесса обжига, где отсутствуют мелкие детали. Таким образом, для изображения имеем $M \times N$ областей, $M = 40$, $N = 30$. Для каждой из $D = 1200$ областей вычисляется входной вектор текстурных характеристик X^i, $i = 1, 2, ..., D$, который содержит составляющие $X^i = [t_1^i, t_2^i, ..., t_{11}^i]^{\mathrm{T}}$, где t_j^i соответствует текстурной характеристике с номером j. Тем самым формируется входное множество векторов $I_X = \{X^1, X^2, ..., X^D\}$. Из данного множества входных векторов необходимо выделить четыре сегмента (класса): "факел", "материал", "футеровка" и "корпус". В результате сегментации на основе входного множества векторов I_X необходимо получить множество номеров сегментов $I_c = \{c_1, c_2, ..., c_D\}$, где c_i — номер сегмента для вектора X^i, $i = 1, 2, ..., D$. Затем необходимо оценить качество сегментации путем сравнения полученной сегментации I_c с эталонным множеством номеров сегментов $E_t = \{s_1, s_2, ..., s_D\}$, где s_i — эталонный номер сегмента для вектора X^i,

$i = 1, 2, ..., D$. Множество E_t задается экспертом. За меру различия сегментов принимается средняя ошибка сегментации $E_{\text{ср}}$, которая равна нулю, если пары номеров сегментов c_i и s_i полностью совпадают, и единице, если c_i и s_i полностью отличаются.

Заранее неизвестно, какие из 11 текстурных характеристик, приведенных выше, лучше всего отражают текстурные особенности изображения процесса обжига. Необходимо найти такой набор текстурных характеристик размерности $1 \leqslant N \leqslant 11$, который обеспечивает минимум средней ошибки сегментации $E_{\text{ср}}$:

$$E_{\text{ср}} = \min_N / 1 \leqslant N \leqslant 11. \qquad (1)$$

Оценка качества сегментации

Для оценки качества сегментации предлагается следующий подход. С помощью специально разработанного редактора сегментов эксперт формирует эталонное множество номеров сегментов изображения E_t. Изображение с выделенными эталонными сегментами представлено на рис. 1, *б*. Для сравнения полученной сегментации I_c с эталонной E_t строится матрица U размерности $R \times R$, где R — число сегментов (классов), в частности, для рассматриваемой задачи $R = 4$. Элементом матрицы U_{ij} является количество пар номеров сегментов, входящих в множества I_c и E_t, которые равны соответственно $c_n = i$ и $s_n = j$:

$$U_{ij} = \sum_{n=1}^{D} I(n, i, j),$$

где $i, j = 1, ..., R$, $I(n, i, j)$ — функция вида

$$I(n, i, j) = \begin{cases} 1, \text{ если } (s_n = i) \wedge (c_n = j), \\ 0, \text{ если иначе,} \end{cases}$$

где s_n и c_n — соответственно полученный и эталонный номер сегмента для области $n = 1, 2, ..., D$, входящие во множества I_c и E_t.

Затем для каждого из классов k вычисляется степень совпадения сегментов $P_{\text{ср}k}$ как отношение

$$P_{\text{ср}k} = U_{kk} / \left(\sum_{i=1}^{R} U_{ki} + \sum_{i=1}^{R} U_{ik} - U_{kk} \right).$$

Данное выражение эквивалентно отношению площади пересечения на изображении полученного сегмента и эталонного к суммарной площади этих сегментов за вычетом площади пересечения (входит в сумму дважды). Степень совпадения $P_{\text{ср}k}$ равна единице, если сегменты полностью совпадают (все элементы матрицы, кроме диагональных, равны нулю), и нулю, если сегменты полностью отличаются (все диагональные элементы равны нулю). В качестве альтернативной величины принята ошибка сегментации $E_{\text{ср}k}$ для сегмента k, т. е. степень несовпадения:

$$E_{\text{ср}k} = 1 - P_{\text{ср}k}.$$

Также вычисляются средняя степень совпадений $P_{\text{ср}}$ и средняя ошибка $E_{\text{ср}}$:

$$P_{\text{ср}} = \sum_{k=1}^{R} P_{\text{ср}k} / R;$$
$$E_{\text{ср}} = 1 - P_{\text{ср}}. \qquad (2)$$

Средний процент ошибки вычисляется как $E_{\text{ср}} \cdot 100\ \%$.

Сегментация изображения с применением самоорганизующихся карт

Сегментация изображения может осуществляться как методом k-средних, так и с помощью математического аппарата самоорганизующихся карт — одной из разновидностей искусственных нейронных сетей. Обучение самоорганизующейся карты осуществляется на основании алгоритма SOM, предложенного Т. Кохоненом [2], с рядом модификаций. Алгоритм включает в себя этапы инициализации сети, выборки входного вектора, поиска максимального подобия, коррекции. Повторение этапов осуществляется до тех пор, пока изменения весов сети станут незначительны. На этапе выборки входной вектор X выбирается не случайно из входного множества векторов, а на каждой итерации задается следующий по порядку вектор из входного множества. Вместо Евклидова расстояния, обычно используемого в алгоритме SOM, применяется расстояние по Камберру, которое позволяет учесть существенные отличия абсолютных значений компонент (текстурных характеристик) входных векторов. Расстояние по Камберру между векторами A и B размерности N имеет вид

$$L(A, B) = \sum_{i=1}^{R} \frac{|A_i - B_i|}{|A_i + B_i|}. \qquad (3)$$

Поиск победившего нейрона i осуществляется в соответствии с формулой (4):

$$i = \arg\min_j L(X, w_j), \qquad (4)$$

где $j = 1, 2, ..., V$ — общее число нейронов в решетке.

Обучение самоорганизующейся карты осуществляется за счет итерационной коррекции весов нейронов. На шаге n коррекция веса нейрона j осуществляется по формуле

$$w_j(n + 1) = w_j(n) + \eta(n) h_{j,\, i}(n)(X - w_j(n)),$$

где $\eta(n)$ — параметр скорости обучения; $h_{j,\, i}(n)$ — функция окрестности с центром в победившем нейроне i.

Параметр скорости обучения изменяется в соответствии с формулой

$$\eta(n) = \eta_0 \exp\left(-\frac{n}{\tau_2}\right),$$

где η_0 — начальное значение скорости обучения; τ_2 — временная константа.

Функция окрестности находится как

$$h_{j,\,i}(n) = \exp\left(-\frac{d_{j,\,i}^2}{2\sigma^2(n)}\right),$$

где $\sigma(n)$ — эффективная ширина топологической окрестности; $d_{j,\,i}$ — латеральное расстояние между j-м нейроном и победившим нейроном i,

$$\sigma(n) = \sigma_0 \exp\left(-\frac{n}{\tau_1}\right),$$

где σ_0 — начальное значение эффективной ширины; τ_1 — временная константа [5].

После построения самоорганизующейся карты необходимо разбить ее на R классов. Для этой задачи можно применять ряд методов, в том числе метод k-средних [2]. Однако так как имеется эталонная сегментация изображения, можно осуществить классификацию самоорганизующейся карты, которая обеспечивает высокую степень совпадения с эталоном.

Каждому из нейронов $i = 1, 2, ..., V$ поставлен в соответствие вектор $Y_i = \{y_{i1}, y_{i2}, ..., y_{iR}\}$, составляющая которого y_{ij}, $j = 1, 2, ..., R$, представляет собой количество наиболее близких векторов текстурных характеристик эталонного изображения, относящихся к классу (сегменту) с индексом j. Составляющие векторов N_i определяют следующим образом. Для каждого вектора текстурных характеристик X^l, $l = 1, 2, ..., D$, эталонного изображения в соответствии с мерой (3) по формуле (4) вычисляется наиболее близкий нейрон i самоорганизующейся карты. Затем в соответствующем i-му нейрону векторе Y_i на единицу увеличивается составляющая y_{ij} с индексом j, равным эталонному номеру сегмента s_l для вектора X^l. Другими словами

$$y_{is_l} = y_{is_l} + 1.$$

После проверки всех векторов текстурных характеристик X^l, $l = 1, 2, ..., D$, номер сегмента для каждого из нейронов $i = 1, 2, ..., V$ вычисляется как

$$c_i = \arg\max_j y_{ij}. \qquad (5)$$

Вычислительный эксперимент показал, что для случая с четырьмя нейронами и выбором из двух классов, описанный выше метод в 99,9 % случаев обеспечивает повышение средней степени совпадений $P_{\text{ср}}$ (2) с эталоном. При этом, если для одного из нейронов выбор номера сегмента по формуле (5) приводит к уменьшению средней степени совпадений $P_{\text{ср}}$ (0,1 % случаев), то это уменьшение в среднем составляет менее 0,02 усл. ед., что является приемлемым для рассматриваемой задачи.

Для задачи построения самоорганизующейся карты на основании векторов текстурных характеристик изображения процесса обжига параметры обучения карты выбраны следующим образом: число нейронов 8 × 8 ($V = 64$), начальное значение скорости обучения $\eta_0 = 0{,}1$, временная константа скорости обучения $\tau_2 = 1000$, начальное значение эффективной ширины топологической окрестности нейрона-победителя $\sigma_0 = 1{,}5$, временная константа функции окрестности $\tau_1 = 2000$ [4].

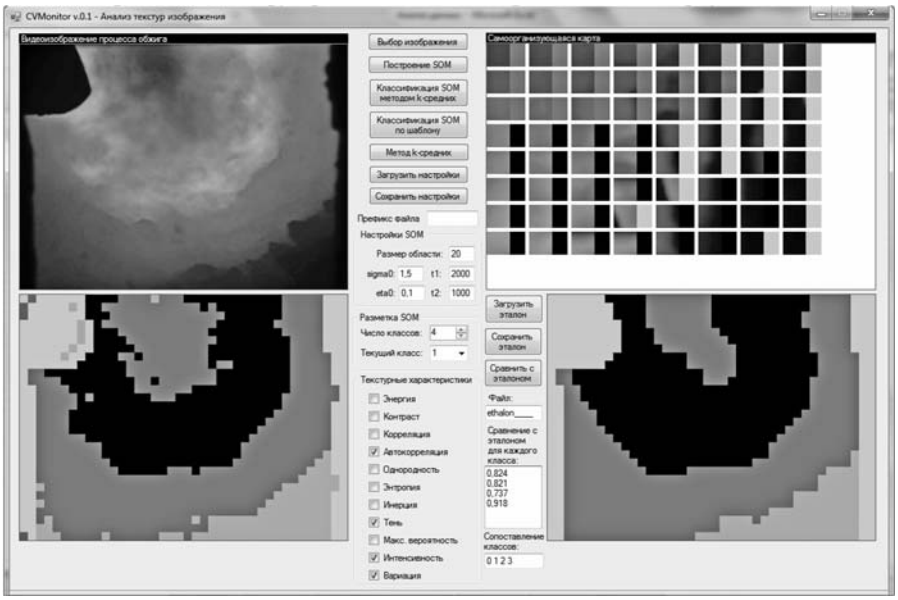


Рис. 2. Экранная форма программы для исследования различных методов сегментации видеоизображений на основе текстурных характеристик

Исследование методов сегментации

Для исследования различных методов сегментации видеоизображений на основе текстурных характеристик разработано программное обеспечение, экранная форма которого представлена на рис. 2. Программа позволяет загрузить из файла изображение, в данном случае выбрано видеоизображение процесса обжига во вращающейся печи (рис. 2, слева вверху). В параметрах настройки задается размер областей, на которые разбивается изображение (в рассматриваемом случае 20 пикселей), возможен выбор любой комбинации из 11 текстурных характеристик для анализа этих областей. Задается число классов (сегмен-

тов), которые могут быть найдены как методом *k*-средних, так и с построением самоорганизующейся карты (SOM).

Результат сегментации выводится под исходным изображением (рис. 2, слева внизу). Построенная самоорганизующаяся карта выводится в виде решетки нейронов 8 × 8 (рис. 2, справа вверху), каждому из которых соответствуют наиболее близкие области изображения, а также номер класса, отображаемый цветом от черного до светло-серого (для случая с четырьмя классами выбраны четыре градации цвета: черный, темно-серый, серый и светло-серый). Номера классов для каждого нейрона SOM можно задавать как вручную (с помощью клика мыши), так и с помощью автоматической классификации SOM методом *k*-средних и классификации по эталону. Есть возможность сохранять/загружать из файла настройки весов самоорганизующейся карты и результат разбиения ее на классы, что позволяет использовать эти файлы для приложения, распознающего изображения технологического процесса в реальном времени. При этом не требуется осуществлять классификацию самоорганизующейся карты заново.

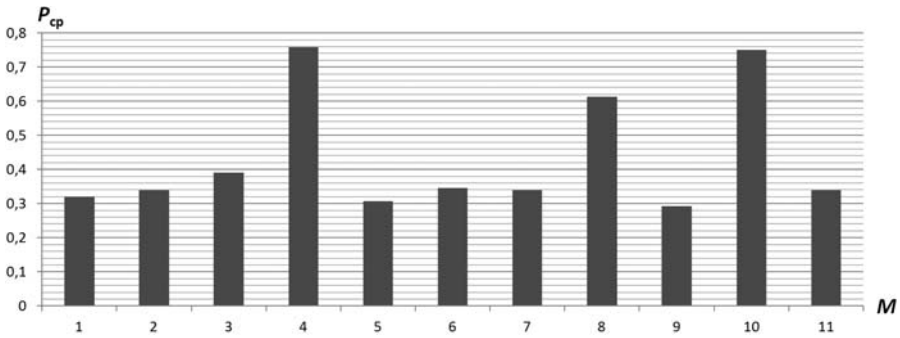


Рис. 3. Средняя степень совпадений $P_{ср}$ сегментации для каждой из 11 текстурных характеристик, M — номер текстурной характеристики

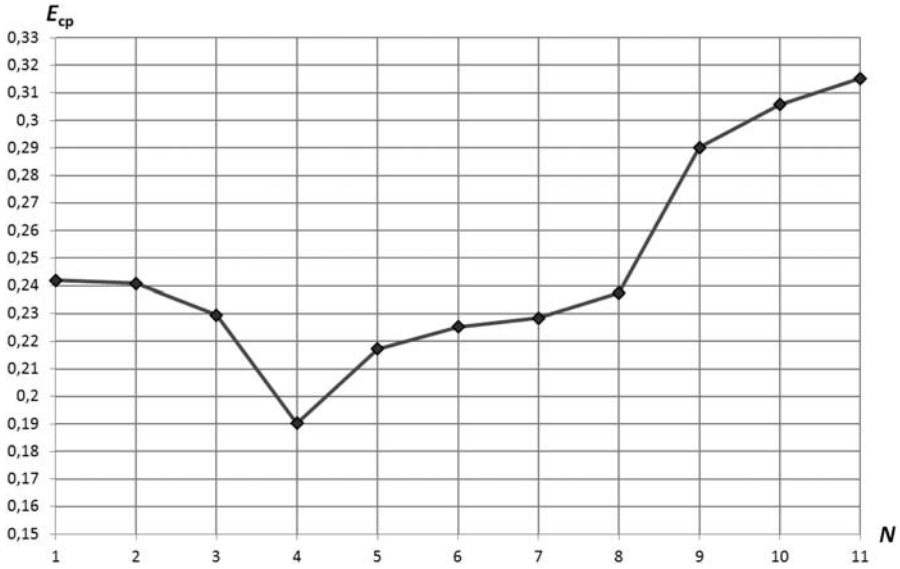


Рис. 4. Средняя ошибка сегментации $E_{ср}$ векторов текстурных характеристик размерности N

Для оценки результатов сегментации реализовано редактирование, сохранение и загрузка из файла эталонной сегментации изображения (рис. 2, справа внизу), здесь же рассчитывается степень совпадения для каждого из сегментов полученной сегментации и эталонной.

С помощью разработанной программы в первую очередь осуществлен подбор комбинации текстурных характеристик, которые дают наименьшую среднюю ошибку сегментации $E_{ср}$. Изначально проанализирована средняя степень совпадений сегментации $P_{ср}$ для каждой из $M = 1, 2, ..., 11$ текстурных характеристик методом самоорганизующихся карт с классификацией по эталону (рис. 3).

Затем исследованы средние ошибки сегментации $E_{ср}$ в зависимости от различных наборов текстурных характеристик размерности N (рис. 4). Размерности $N = 1$ соответствует составляющая автокорреляция; $N = 2$ — автокорреляция и интенсивность; $N = 3$ — автокорреляция, тень и интенсивность; $N = 4$ — автокорреляция, тень, интенсивность и вариация; $N = 5$ — автокорреляция, однородность, тень, интенсивность и вариация; $N = 6$ — корреляция, автокорреляция, однородность, тень, интенсивность и вариация; $N = 7$ — контраст, корреляция, автокорреляция, однородность, тень, интенсивность и вариация; $N = 8$ — контраст, корреляция, автокорреляция, однородность, энтропия, тень, интенсивность и вариация; $N = 9$ — те же, что и при $N = 8$, и инерция; $N = 10$ — те же, что и при $N = 9$, и энергия; $N = 11$ — все 11 текстурных характеристик.

В соответствии с (1) минимальная средняя ошибка сегментации $E_{ср}$ достигается при размерности набора текстурных характеристик, равной четырем (составляющие: автокорреляция, тень, интенсивность, вариация). Таким образом, найден квазиоптимальный набор текстурных характеристик, обеспечивающий минимальное значение средней ошибки $E_{ср}$. Полным перебором всех возможных комбинаций наборов текстурных характеристик может быть найдена оптимальная комбинация, однако это существенно усложняет задачу.

Время расчета текстурных характеристик составляет 130 мс, что является приемлемым для системы технического зрения, которая работает в реальном времени и требует распознавания изображения в печи обжига через каждые 1—2 с. Время расчета приведено для компьютера со следующими

Сравнение методов сегментации видеоизображения процесса обжига по приведенным показателям

Показатель	Метод *k*-средних	Метод SOM с автоматической классификацией методом *k*-средних	Метод SOM с классификацией по эталону
Совпадение сегмента "Факел" с эталонным, усл. ед.	0,861	0,846	0,826
Совпадение сегмента "Материал" с эталонным, усл. ед.	0,785	0,755	0,817
Совпадение сегмента "Футеровка" с эталонным, усл. ед.	0,396	0,388	0,748
Совпадение сегмента "Корпус" с эталонным, усл. ед.	0,459	0,37	0,848
Средняя степень совпадений сегментации, усл. ед.	0,6253	0,5898	0,80975
Средняя ошибка сегментации, усл. ед.	0,3747	0,4102	0,19025
Средняя ошибка сегментации, %	37,47	41,02	19,03

характеристиками: процессор Intel Core i5 с частотой 2,7 ГГц (четыре ядра), оперативная память типа DDR3 объемом 4 Гбайт с частотой 1333 МГц.

Для вектора текстурных характеристик с четырьмя составляющими (автокорреляция, тень, интенсивность, вариация) проведено сравнение различных методов сегментации изображения (см. таблицу).

Метод построения SOM с классификацией по эталону для видеоизображения процесса обжига имеет среднюю ошибку сегментации почти в 2 раза меньше, чем метод *k*-средних и метод построения SOM с автоматической классификацией методом *k*-средних. Это обеспечивается одним из достоинств самоорганизующихся карт — возможностью непосредственного доступа к каждому нейрону, что позволяет выполнить автоматическое разбиение на классы в целях уменьшения ошибки между полученной и эталонной сегментацией изображения.

Заключение

Разработанный метод сегментации изображений с применением текстурного анализа на основе самоорганизующихся карт с классификацией по эталону используется для анализа изображений процесса обжига во вращающихся печах цементных производств, в частности, в печи № 1 ЗАО "Оскол-цемент" (г. Старый Оскол). На основе проведенной сегментации изображения находятся выходные образы, по которым операторы печи осуществляют мониторинг и оперативное управление печью. Также метод может быть с успехом применен для анализа изображений на других промышленных объектах, в частности, керамзитовых печах.

Работа выполнена в рамках гранта № А-27/12 "Программы стратегического развития БГТУ им. В. Г. Шухова на 2012—2016 гг." (№ 2011-ПР-146), гранта РФФИ № 12-07-97526-р_центр_а.

Список литературы

1. **Классен В. К., Нусс М. В.** Оптимизация обжига цементного клинкера с использованием информационных технологий // Известия высших учебных заведений. Строительство. 2008. № 1. С. 34—40.

2. **Кохонен Т.** Самоорганизующиеся карты: пер. 3-го англ. изд. М.: БИНОМ. Лаборатория знаний, 2010. 655 с.

3. **Патана Е. И.** Статистический анализ и кластеризация основных текстурных функционалов // Известия Южного федерального университета. Технические науки. 2008. Т. 81, № 4. С. 192—198.

4. **Мицель А. А., Колодовникова Н. В., Протасов К. Т.** Непараметрический алгоритм текстурного анализа аэрокосмических снимков // Известия Томского политехнического университета. Технические науки. 2005. Т. 308, № 1. С. 65—70.

5. **Haykin S.** Neural Networks and Learning Machines. 3rd ed. Prentice Hall, 2009. 906 p.

CONTENTS

An algorithm is presented for creating adaptive grid based on triangular elements, that is intended for use in an effective algorithm for object recognition, motion detection and video compression, which allows to achieve a relatively high quality of the encoded image (in lossy image compression). The characteristics of the proposed algorithm in terms of image compression, control of density grid are investigated.

Keywords: triangulation, adaptive grid, image and video compression

Materials of researches on recognition of signals with the close time-frequency structure are represented. It is offered to use as recognition signs the arranged sequences them wavelet coefficients. Feasibility of synthesis wavelet coefficients on the basis of frame transforms is justified. Results of computer simulation are analyzed.

Keywords: recognition of signals, vector of signs, frame transforms

The Analysis for the Economic Development Dynamics of the USA, the UK, Japan and the Russian Federation from 1998 to 2011 by the Main Stock Indexes on the Basis of New Information Approach to Stochastic Processes Analysis is carried out. The Tables of the Indexes Information Mismatch in Elaboration and in Comparison with Each Other and the Respective Summary of the Economical Development Dynamics are given. The Forecast of the Indexes Further Dynamics to 2014 is adduced.

Keywords: economic development dynamics, stock indexes, autoregressive observation model, whitening filter method, information mismatch, forecasting

The complex index of determination of quality (accuracy) of operation of the neural networks, considering values of the functional tolerances is offered. The description of properties of a complex index and its use on an example of the neural networks implementing computation of basic mathematical functions is provided. Dependence of quality of operation of neural networks on the selected function of training is shown.

Keywords: neural networks, accuracy of operation, quality of operation of a neural network

The approach to increase of efficiency of decision-making of tax regulation at the expense of use of simulation by Kokhonen's neural networks is considered. Within this approach the method of selection of factors of an assessment of a financial status of the enterprises in procedure of their clustering on neural networks on the basis of a Bayesian approach is offered. Results of researches of efficiency of the offered method on an example of an assessment of solvency of the enterprises in different branches of a national economy are provided.

Keywords: neural networks, tax administration, clustering, selection of factors, Bayesian approach

Paper considers developed method of image segmentation of firing process with texture analysis based on self-organizing maps. It shows a software implementation of the proposed method, illustrates the functionality of the developed software package. It describes finding of a set of texture features that provide the quasi-optimal image segmentation of the firing process. It gives the comparison of image segmentation based on self-organizing maps with *k*-means method. It shows the advantages of using the apparatus of self-organizing maps for classification of texture characteristics vectors.

Keywords: image segmentation, firing process, rotary kiln, self-organizing maps, neural networks, texture analysis, software package

Адрес редакции:

107076, Москва, Стромынский пер., 4

Телефон редакции журнала **(499) 269-5510**

E-mail: it@novtex.ru

Дизайнер *Т.Н. Погорелова*. Технический редактор *Е. В. Конова*.

Корректор *Е. В. Комиссарова*.

Сдано в набор 05.03.2013. Подписано в печать 22.04.2013. Формат 60×88 1/8. Бумага офсетная.

Усл. печ. л. 8,86. Заказ IT513. Цена договорная.

Журнал зарегистрирован в Министерстве Российской Федерации по делам печати, телерадиовещания и средств массовых коммуникаций.

Свидетельство о регистрации ПИ № 77-15565 от 02 июня 2003 г.

Оригинал-макет ООО "Адвансед солюшнз". Отпечатано в ООО "Адвансед солюшнз".

105120, г. Москва, ул. Нижняя Сыромятническая, д. 5/7, стр. 2, офис 2.