

ПОИСК ВЕСОВ В ЗАДАЧЕ ВЗВЕШЕННОЙ АППРОКСИМАЦИИ ВРЕМЕННЫМ РЯДОМ КОНЕЧНОГО РАНГА*

Н. К. Звонарев

Санкт-Петербургский государственный университет,
Российская Федерация, 199034, Санкт-Петербург, Университетская наб., 7–9

Рассматривается задача взвешенной аппроксимации временного ряда рядом конечного ранга с целью оценивания сигнала. Решается проблема поиска весов, которые приводят к улучшению точности оценок. Строится и теоретически обосновывается эффективный метод численного расчета поиска весов с помощью теории квадратичной оптимизации. Для того чтобы получить эффективный алгоритм, задача квадратичной оптимизации с большим числом ограничений сводится к последовательности задач с меньшим числом ограничений и критерии останова. Для обоснования алгоритма доказывается эквивалентность различных формулировок исходной задачи оптимизации. Проводится численное моделирование, подтверждающее эффективность алгоритма и улучшение точности метода оценивания сигнала. Библиогр. 10 назв. Ил. 2. Табл. 1.

Ключевые слова: временные ряды, метод Cadzow, ряды конечного ранга, взвешенный метод наименьших квадратов, сингулярное разложение, квадратичное программирование.

Введение. Рассмотрим задачу извлечения сигнала $\mathbb{S} = (s_1, \dots, s_N)$ из зашумленного временного ряда $\mathbb{X} = \mathbb{S} + \mathbb{N}$, где \mathbb{S} управляется *линейной рекуррентной формулой* (ЛРФ) порядка r : $s_n = \sum_{i=1}^r a_i s_{n-i}$, $n = r + 1, \dots, N$, $a_r \neq 0$, \mathbb{N} – белый шум. Ряды, управляемые ЛРФ, могут быть записаны в параметрическом виде $s_n = \sum_i P_i(n) \exp(\alpha_i n) \cos(2\pi\omega_i n + \psi_i)$, где $P_i(n)$ – многочлены от n [1].

Известно, что для выделения такого рода сигналов хорошо работают методы, основанные на оценке сигнального подпространства (subspace-based methods) и, в частности, анализ сингулярного спектра (см. [1–4] и [5, глава 4]). Идея методов состоит в следующем: зафиксируем *длину окна* L , $1 < L < N$, положим $K = N - L + 1$ и определим *траекторную матрицу* для ряда $\mathbb{S} \in \mathbb{X}_N$ (\mathbb{X}_N – множество вещественных временных рядов длины N):

$$\mathbf{S} = \mathcal{T}_L(\mathbb{S}) = \begin{pmatrix} s_1 & s_2 & \dots & s_K \\ s_2 & s_3 & \dots & s_{K+1} \\ \vdots & \vdots & \ddots & \vdots \\ s_L & s_{L+1} & \dots & s_N \end{pmatrix},$$

где \mathcal{T}_L обозначает биекцию между \mathbb{X}_N и \mathcal{H} , и \mathcal{H} – множество ганкелевых матриц порядка $L \times K$ с одинаковыми значениями на побочных диагоналях $i + j = \text{const}$.

Скажем, что ряд \mathbb{S} , траекторная матрица \mathbf{S} которого имеет неполный ранг r , называется *рядом L -ранга r* . Если L -ранг не зависит от L при достаточно больших N и L , ряд называется *рядом конечного ранга*. Рассмотрим задачу аппроксимации временного ряда рядом L -ранга по взвешенной норме [6, 7]:

$$\sum_{i=1}^N q_i (x_i - y_i)^2 \rightarrow \min_{\mathbb{Y}: \text{rank } \mathbf{Y} = \mathcal{T}_L(\mathbb{Y}) \leq r}, \quad (1)$$

*Работа выполнена при финансовой поддержке РФФИ (грант № 16-04-00821).

© Санкт-Петербургский государственный университет, 2016

где $\mathbb{X} = (x_1, \dots, x_N) \in \mathbb{X}_N$ — исходный временной ряд, $\mathbb{Y} = (y_1, \dots, y_N)$ — требуемая аппроксимация, $\mathbf{Y} = \mathcal{T}_L(\mathbb{Y}) = \mathcal{T}(\mathbb{Y})$ — соответствующая траекторная матрица, $q_i > 0$, $i = 1, \dots, N$, — веса (в дальнейшем называемые *весами ряда*).

Полученное решение \mathbb{Y}^* задачи (1) можно использовать как оценку сигнала \mathbb{S} рядом конечного ранга. Более того, полученная оценка позволяет решить множество связанных задач: оценка коэффициентов ЛРФ и параметров ряда, прогнозирование ряда, заполнение пропусков и так далее [1]. Заметим, что в случае белого шума естественно выбрать веса q_i равными.

Обозначим соответствующее скалярное произведение в \mathbb{X}_N как $\langle \mathbb{X}, \mathbb{Y} \rangle_q = \sum_{i=1}^N q_i x_i y_i$. Для решения задачи (1) в [6] рассматривается эквивалентная задача структурной аппроксимации матрицей неполного ранга (Structured Low-Rank Approximation, SLRA):

$$\|\mathbf{X} - \mathbf{Y}\|_{\mathbf{C}}^2 \rightarrow \min_{\substack{\text{rank } \mathbf{Y} \leq r \\ \mathbf{Y} \in \mathcal{H}}}, \quad (2)$$

где $\mathbf{X} = \mathcal{T}(\mathbb{X})$, а матричная норма $\|\cdot\|_{\mathbf{C}}$ (\mathbf{C} — диагональная, положительно определенная матрица) порождена следующим скалярным произведением:

$$\langle \mathbf{Y}, \mathbf{Z} \rangle_{\mathbf{C}} = \text{tr}(\mathbf{Y}\mathbf{C}\mathbf{Z}^T) = \sum_{l=1}^L \sum_{k=1}^K c_k y_{l,k} z_{l,k},$$

где $\mathbf{C} = \text{diag}(c_1, \dots, c_K)$, $\mathbf{Y} = (y_{l,k})$, $\mathbf{Z} = (z_{l,k})$. Между весами q_i в задаче (1) аппроксимации ряда и весами c_i в задаче (2) (в дальнейшем называемые *матричными весами*) существует соотношение [6, предл. 4] для эквивалентности норм.

Основная проблема в [6] состояла в нахождении весов c_i , соответствующих равным весам q_i , естественных для модели с белым шумом. Тем самым решение задачи (2) соответствовало оценке ряда, являющейся решением задачи невзвешенных наименьших квадратов в пространстве рядов. Однако опять же согласно [7] и [8], если это и возможно, то только при вырожденной матрице \mathbf{C} , что недопустимо при решении задачи методами типа Oblique Cadzow [6, замечание 4]. Следовательно, равномерностью весов ряда q_i приходится жертвовать и использовать матричные веса c_i такие, чтобы \mathbf{C} была невырожденной, а веса q_i лишь близкими к равномерным.

Цель статьи — разработка эффективного алгоритма поиска весов при зафиксированной мере невырожденности матрицы \mathbf{C} . В качестве такой меры ограничим число обусловленности \mathbf{C} снизу. Так же как и в [6], введем параметр $0 < \alpha \leq 1$ и потребуем, чтобы все c_i удовлетворяли неравенствам $c_i \geq 0$ и

$$\frac{\min_i c_i}{\max_i c_i} \geq \alpha, \quad (3)$$

что эквивалентно неравенству $\text{Cond}(\mathbf{C}) \leq 1/\alpha$. После этого поставим задачу аппроксимации единичных весов ряда по наиболее естественной евклидовой норме. В этом и состоит ключевое отличие от предыдущего подхода, описанного в [6, предл. 5] и неприменимого в большинстве практических ситуаций. Например, в [6] требуется, чтобы длина ряда N была кратной длине окна L . Данная работа посвящена решению задачи в такой постановке, что требует применение теории экстремальных задач и квадратичного программирования (КП). Поэтому получение эффективного по скорости работы алгоритма поиска весов является нетривиальной задачей.

В [6, раздел 5] показано, что чем равномернее веса ряда в задаче (1), тем точнее получается решение. Более того, как было показано в [6], параметр α влияет на скорость сходимости метода Oblique Cadzow: чем больше $\text{Cond}(\mathbf{C})$ (что равнозначно меньшему значению α), тем медленнее сходится метод. В разделе 6 показано, что эти соотношения выполняются и для весов, полученных новым методом.

1. Постановка задачи аппроксимации весов. Для удобства переформулируем предложение [6, предл. 4], связывающее c_i и q_i , в матричном виде. Для этого рассмотрим матрицу $\mathbf{B} = (b_{i,j})$ порядка $N \times K$, имеющую следующий вид:

$$b_{i,j} = \begin{cases} 1, & \text{для } i = j, \dots, j + L - 1, \\ 0, & \text{в противном случае.} \end{cases} \quad (4)$$

Заметим, что \mathbf{B} — матрица полного ранга. Предложение выглядит следующим образом.

Предложение 1. Пусть $\mathbf{Y} = \mathcal{T}(\mathbb{Y})$, $\mathbf{Z} = \mathcal{T}(\mathbb{Z})$, $Q = (q_1, \dots, q_N)^T$, $C = (c_1, \dots, c_K)^T$, $\mathbf{C} = \text{diag}(C)$. Тогда $\langle \mathbb{Y}, \mathbb{Z} \rangle_q = \langle \mathbf{Y}, \mathbf{Z} \rangle_{\mathbf{C}}$ для любых $\mathbb{Y}, \mathbb{Z} \in \mathbf{X}_n$ тогда и только тогда, когда $Q = \mathbf{B}C$.

В общем виде задачу аппроксимации весов сформулируем следующим образом.

Задача 1.

$$\varphi^* = \min_{C \in \mathcal{C}} \varphi(C), \quad \text{где } \varphi(C) = \|\mathbf{B}C - \mathbb{1}_N\|^2, \quad (5a)$$

$$\mathcal{C} = \left\{ C \mid c_i \geq 0, \quad i = 1, \dots, K, \quad \frac{\min_i c_i}{\max_i c_i} \geq \alpha \right\}, \quad (5b)$$

где $\mathbb{1}_N = (1, \dots, 1)^T \in \mathbb{R}^N$ — требуемые веса ряда (вектор из N единиц), $0 < \alpha \leq 1$ — параметр, регулирующий степень вырожденности матрицы \mathbf{C} , $\|\cdot\|$ — евклидова норма в \mathbb{R}^N .

Замечание 1. В дальнейшем случай $\alpha = 1$ опустим. При $\alpha = 1$ все $c_i = c$ равны между собой, и задача 1 сводится к поиску множителя c такого, что $\|c\mathbf{B}\mathbb{1}_K - \mathbb{1}_N\|^2$ достигает своего минимума. Он достигается в точке $c = \mathbb{1}_K^T \mathbf{B}^T \mathbb{1}_N / (\mathbb{1}_K^T \mathbf{B}^T \mathbf{B} \mathbb{1}_K)$.

Решение задачи 1 можно использовать для решения задачи (2) и тем самым для решения задачи (1).

2. Эквивалентные формулировки. Рассмотрим выражение $\varphi(C)/2$, где $\varphi(C)$ определено в (5a). Преобразуем его, не учитывая константное слагаемое, а множество \mathcal{C} из (5b) запишем в виде набора линейных ограничений. Получим следующую эквивалентную формулировку.

Задача 2.

$$f^* = \min_{C \in \mathcal{C}} f(C), \quad \text{где } f(C) = \frac{1}{2} C^T \mathbf{S} C - L \mathbb{1}_K^T C, \quad (6a)$$

$$\mathcal{C} = \{ C \mid c_i - \alpha c_j \geq 0, \quad i \neq j, \quad 1 \leq i, j \leq K \}, \quad (6b)$$

где $\mathbf{S} = (s_{i,j}) = \mathbf{B}^T \mathbf{B}$ — положительно определенная матрица размера $K \times K$ с элементами

$$s_{i,j} = \begin{cases} L - |i - j|, & |i - j| \leq L, \\ 0, & \text{в противном случае,} \end{cases} \quad (7)$$

а вектор $L\mathbf{1}_K = \mathbf{B}^T \mathbf{1}_N = (L, \dots, L)^T \in \mathbb{R}^K$.

Теперь задача (2) представлена в том виде, для которого разработана теория квадратичного программирования [9, с. 111]. Справедливо следующее утверждение.

Предложение 2.

1. Задача 2 имеет единственное решение C^* .

2. Ее решение $C^* = (c_1^*, \dots, c_K^*)^T$ является палиндромом, то есть для любого индекса $1 \leq i \leq K$ справедливо $c_i^* = c_{K-i+1}^*$.

ДОКАЗАТЕЛЬСТВО.

1. Задача 2 — задача КП с набором из K^2 линейных ограничений (6b) и целевой функцией (6a), квадратичная форма в которой положительно определена, поэтому решение такой задачи единственно [10, с. 452].

2. Достаточно рассмотреть вектор $C^{**} = (c_K^*, \dots, c_1^*)^T$ и заметить, что $f(C^*) = f(C^{**})$ и $C^{**} \in \mathcal{C}$ в (6b); значит, $C^* = C^{**}$, что и требовалось доказать. ■

Для решения задачи 2 можно использовать методы квадратичного программирования, но, к сожалению, K^2 линейных ограничений являются серьезной проблемой при решении задачи на практике. Рассмотрим еще две эквивалентные формулировки, свойства которых используются при решении задачи.

Сделаем дополнительное предположение: пусть в точке j веса достигают своего максимума (тем самым снизим число линейных ограничений до линейного по K размера), а само j будем перебирать в цикле от 1 до $\lceil K/2 \rceil$. При этом воспользуемся тем фактом, что решение является палиндромом. Формально задача выглядит следующим образом.

Задача 3.

$$f^{**} = \min_{j=1, \dots, \lceil K/2 \rceil} f_j^*, \quad (8a)$$

$$f_j^* = \min_{C \in \mathcal{C}_j} f(C), \quad (8b)$$

$$\mathcal{C}_j = \{C \mid c_i = c_{K-i+1}, i = 1, \dots, \lceil K/2 \rceil\}; \quad (8c)$$

$$c_i - \alpha c_j \geq 0, i = 1, \dots, j-1, j+1, \dots, \lceil K/2 \rceil; \quad (8d)$$

$$c_j - c_i \geq 0, i = 1, \dots, j-1, j+1, \dots, \lceil K/2 \rceil\}. \quad (8e)$$

Чтобы не перебирать все возможные j , удобно уметь проверять, дает ли полученное на очередной итерации решение глобальный минимум. Для такой проверки рассмотрим еще одну эквивалентную формулировку задачи 1. Введем дополнительную переменную c_{\max} , хранящую максимальный вес, и перейдем к новым переменным $\hat{c}_i = c_{\max} - c_i$, $i = 1, \dots, K$, определяющим разницу между максимальным среди всех и текущим весами. При этом выполняются неравенства $c_{\max} \geq 0$, $\hat{c}_i \geq 0$. Между векторами $C = (c_1, \dots, c_K)^T$ и $\hat{C} = (\hat{c}_1, \dots, \hat{c}_K, c_{\max})^T$ существует простое линейное соответствие: $C = \mathbf{H}\hat{C}$, где $\mathbf{H} \in \mathbb{R}^{K \times (K+1)}$ — матрица следующего вида:

$$\mathbf{H} = \begin{pmatrix} -1 & & 0 & 1 \\ & \ddots & & \vdots \\ 0 & & -1 & 1 \end{pmatrix}. \quad (9)$$

Условие (3), устанавливающее границу снизу для весов, в новых обозначениях записывается как $(1 - \alpha)c_{\max} - \hat{c}_i \geq 0, i = 1, \dots, K$.

Получаем следующую задачу квадратичного программирования с линейными ограничениями, но с нестрогой выпуклой целевой функцией.

Задача 4.

$$\hat{f}^* = \min_{\hat{C} \in \hat{\mathcal{C}}} \hat{f}(\hat{C}), \quad \text{где} \quad \hat{f}(\hat{C}) = f(\mathbf{H}\hat{C}) = \frac{1}{2}\hat{C}^T \mathbf{H}^T \mathbf{S} \mathbf{H} \hat{C} - L \mathbf{1}_K^T \mathbf{H} \hat{C}, \quad (10a)$$

$$\hat{\mathcal{C}} = \left\{ \hat{C} \mid \hat{c}_j \geq 0, \quad (1 - \alpha)c_{\max} - \hat{c}_j \geq 0, \quad j = 1, \dots, K \right\}. \quad (10b)$$

Теорема 1. *Задачи 1, 2, 3 и 4 эквивалентны.*

ДОКАЗАТЕЛЬСТВО. Эквивалентность задач 1 и 2 очевидна, так как задача 2 является переформулировкой задачи 1. Для оставшихся задач приведем идею доказательства.

Требуется доказать $f^* \geq f^{**} \geq \hat{f}^* \geq f^*$ и тот факт, что по решению одной из задач можно построить решение любой другой.

Докажем неравенство $f^* \geq f^{**}$. Пусть C^* — решение задачи 2, то есть $C^* = \arg \min_{C \in \mathcal{C}} f(C)$. Положим $j^* = \arg \max_i c_i$. Имеем $C^* \in \mathcal{C}$ и C^* — палиндром, следовательно, $C^* \in \mathcal{C}_{j^*}$. Цепочка неравенств $f^* = f(C^*) \geq f_{j^*}^* = \min_{C \in \mathcal{C}_{j^*}} f(C) \geq \min_{j=1, \dots, \lceil K/2 \rceil} \min_{C \in \mathcal{C}_j} f(C) = f^{**}$ завершает доказательство неравенства.

Докажем неравенство $f^{**} \geq \hat{f}^*$. Пусть $j^* = \arg \min_{j=1, \dots, \lceil K/2 \rceil} f_j^*$, $C_{j^*}^* = \arg \min_{C \in \mathcal{C}_{j^*}} f(C)$ — решение задачи 3. Положим $c_{\max} = c_{j^*, j^*}$, $\hat{c}_i = c_{\max} - c_{j^*, i}$, $i = 1, \dots, K$, где $C_{j^*}^* = (c_{j^*, 1}, \dots, c_{j^*, K})^T$. Нетрудно заметить, что $\hat{C} \in \hat{\mathcal{C}}$, из чего получим $\hat{f}^* \leq \hat{f}(\hat{C}) = f(C_{j^*}^*) = f^{**}$.

Докажем неравенство $\hat{f}^* \geq f^*$. Пусть $\hat{C}^* = \arg \min_{\hat{C} \in \hat{\mathcal{C}}} \hat{f}(\hat{C})$ — решение задачи 4. Положим $c_i = c_{\max} - \hat{c}_i$ и заметим, что $C \in \mathcal{C}$, из чего получим $f^* \leq f(C) = \hat{f}(\hat{C}) = \hat{f}^*$. ■

3. Общий алгоритм решения. Используя эквивалентность, доказанную в теореме 1, можем сформулировать следующий алгоритм решения задачи 1.

Алгоритм 1. Вход: Параметры L, K, α .

Результат: Вектор оптимальных весов C^* .

1. Положить $j = 1$.
2. Решить подзадачу КП $C_j^* = (c_{j,1}, \dots, c_{j,K})^T = \arg \min_{C \in \mathcal{C}_j} f(C)$.
3. Задать эквивалентный вектор \hat{C} , взяв $c_{\max} = c_{j,j}$, $\hat{c}_i = c_{\max} - c_{j,i}$.
4. Проверить, является ли вектор \hat{C} решением задачи 4. Если да, положить $C^* = C_j^*$ решением задачи, иначе взять $j \leftarrow j + 1$ и перейти к пункту 2.

Таким образом, если встретился нужный индекс j , алгоритму не понадобится перебирать оставшиеся индексы и решать подзадачу (8b) лишний раз. Практические эксперименты показывают, что максимальный вес всегда находится на краях: алгоритм 1 останавливается уже при $j = 1$, то есть фактически сводится к задаче КП с положительно определенной квадратичной формой в целевой функции.

Для реализации алгоритма 1 необходимо разработать алгоритмы решения задач из пункта 2 и 4, что и будет сделано в следующих разделах работы. Сначала в разделе 4 рассмотрим пункт 4, а в разделе 5 — пункт 2.

4. Проверка вектора на решение задачи 4. Предложим быстрый алгоритм, проверяющий, является ли заданный вектор \widehat{C} точкой, в которой достигается глобальный минимум в задаче 4, то есть реализующий пункт 4 из алгоритма 1. Для этого применим теорему о необходимом и достаточном условии минимума в задаче квадратичного программирования для задачи 4.

Теорема 2.

1. Рассмотрим вектор $R = (r_1, \dots, r_{K+1})^T = \mathbf{H}^T \mathbf{S} \mathbf{H} \widehat{C} - \mathbf{L} \mathbf{H}^T \mathbf{1}_K$. Тогда \widehat{C} является решением задачи 4 в том и только в том случае, если выполняются условия:

(1a) $R^T \widehat{C} = 0$,

(1b) существует вектор $U = (u_1, \dots, u_K)^T \in \mathbf{R}^K$ такой, что имеют место неравенства $u_i \geq 0$, $i = 1, \dots, K$, $u_i \geq -r_i$, $i = 1, \dots, K$, $(1 - \alpha) \sum_{i=1}^K u_i \leq r_{K+1}$.

2. Положим $u_i = \max(0, -r_i)$, $i = 1, \dots, K$. Тогда \widehat{C} является решением задачи 4 в том и только в том случае, если выполняются условия:

(2a) $R^T \widehat{C} = 0$,

(2b) $(1 - \alpha) \sum_{i=1}^K u_i \leq r_{K+1}$.

ДОКАЗАТЕЛЬСТВО. Первый пункт данной теоремы является переформулировкой теоремы [9, теорема 9.2] для задачи 4, поэтому перейдем сразу к доказательству второго пункта. Первое условие в обоих пунктах одинаковое. Докажем эквивалентность второго условия.

Достаточность. Коэффициенты u_i выбраны так, чтобы условия $u_i \geq 0$ и $u_i \geq -r_i$ выполнялись. Если удовлетворено и второе условие, выполняются все требования первого пункта.

Необходимость. Очевидно, что u_i выбраны наименьшими из всех тех, которые удовлетворяют условиям $u_i \geq 0$ и $u_i \geq -r_i$. Пусть имеем неравенство $(1 - \alpha) \sum_{i=1}^K u_i > r_{K+1}$. Рассмотрим любой вектор $\tilde{U} = (\tilde{u}_1, \dots, \tilde{u}_N)$, удовлетворяющий условию (2b), то есть $(1 - \alpha) \sum_{i=1}^K \tilde{u}_i \leq r_{K+1}$. Тогда существует индекс i такой, что $\tilde{u}_i < u_i$. Следовательно, либо $\tilde{u}_i < 0$, либо $\tilde{u}_i < -r_i$, из чего следует, что не существует такого вектора \tilde{U} , который бы удовлетворял условиям первого пункта. ■

Рассмотрим следующий алгоритм, базирующийся на теореме 2.

Алгоритм 2. Вход: предполагаемое решение \widehat{C} , параметр α .

Результат: Булево значение: является ли \widehat{C} решением задачи 4.

1. Вычислить $R = \mathbf{H}^T \mathbf{S} \mathbf{H} \widehat{C} - \mathbf{L} \mathbf{H}^T \mathbf{1}_K$.

2. Если $R^T \widehat{C} \neq 0$, вернуть FALSE, иначе перейти к следующему пункту.

3. Выбрать в качестве вектора $U = (u_1, \dots, u_K)^T$ следующие значения: $u_i = \max(0, -r_i)$, $i = 1, \dots, K$.

4. Если $(1 - \alpha) \sum_{i=1}^K u_i \leq r_{K+1}$, вернуть TRUE, иначе FALSE.

5. Задача квадратичного программирования специального вида. Теперь переходим к пункту 2 алгоритма 1. Для решения подзадачи (8b) при зафиксированном индексе j воспользуемся так называемым «Primary Active Set»-методом решения задачи КП, описанным в [10]. Специфика задачи позволяет эффективно реализовать алгоритм.

5.1. Общая схема «Active Set»-метода. В общем случае задача квадратичного программирования выглядит следующим образом.

Задача 5.

$$\min_{X \in X} \frac{1}{2} X^T \mathbf{G} X - V^T X, \quad (11a)$$

$$X = \{X \mid A_i^T X = p_i, \quad i \in Y; \quad (11b)$$

$$A_i^T X \geq p_i, \quad i \in Z\}, \quad (11c)$$

где $\mathbf{G} \in \mathbb{R}^{K \times K}$ — произвольная положительно определенная матрица, $V \in \mathbb{R}^K$ — произвольный вектор, Y и Z — множества индексов, векторы $A_i \in \mathbb{R}^K$ вместе с $p_i \in \mathbb{R}$ задают ограничения.

Суть любого «Active Set»-метода состоит в последовательном переборе подмножества ограничений, которые выполняются как равенство для промежуточной точки — кандидата в решение задачи квадратичного программирования. Такое множество называется *рабочим множеством* и обозначается $W \subset Y \cup Z$, при этом всегда $Y \subset W$, а ограничения, лежащие в рабочем множестве, называются *активными*. Ниже представлена схема этого метода для решения задачи 5.

Алгоритм 3 [10, с. 472]. **Вход:** параметры задачи квадратичного программирования: матрица \mathbf{G} , вектор V , множества Y , Z и коэффициенты условий A_i , p_i .

Результат: Решение X^* задачи 5.

1. Найти начальную точку X_0 , удовлетворяющую условиям задачи (11b), (11c), и положить W_0 — множество активных ограничений в этой точке.
2. Положить $k = 0$.
3. Положить $G_k = V - \mathbf{G}X_k$.
4. Решить подзадачу квадратичного программирования и найти множители Лагранжа u_i , $i \in W_k$:

$$P_k^* = \arg \min_{P_k \in P_k} \frac{1}{2} P_k^T \mathbf{G} P_k - G_k^T P_k, \quad (12a)$$

$$P_k = \{P_k \mid A_i^T P_k = 0, \quad i \in W_k\}, \quad (12b)$$

после чего найти u_i из системы уравнений $\sum_{i \in W_k} u_i A_i = \mathbf{G}P_k^* - G_k$.

5. Если $P_k^* = 0$, и все $u_i \geq 0$, $i \in W_k \cap Z$, положить $X^* = X_k$ и STOP.
6. Если $P_k^* = 0$, но существует i такое, что $u_i < 0$, взять $j = \arg \min_{j \in W_k \cap Z} u_j$, положить $W_{k+1} = W_k \setminus \{j\}$, увеличить k на единицу и перейти к п. 3.
7. Положить $\alpha_k = \min \left(1, \min_{i \notin W_k, A_i^T P_k^* < 0} \frac{p_i - A_i^T X_k}{A_i^T P_k^*} \right)$.

8. Положить $X_{k+1} = X_k + \alpha_k P_k^*$.
9. Если $\alpha_k < 1$, положить $j = \arg \min_{i \notin W_k, A_i^T P_k^* < 0} \frac{p_i - A_i^T X_k}{A_i^T P_k^*}$ и $W_{k+1} = W_k \cup \{j\}$, иначе $W_{k+1} = W_k$.
10. Положить $k \leftarrow k + 1$ и перейти к пункту 3.

5.2. Специфика задачи. Подзадача (8b) при фиксированном j переписывается в терминах задачи 5 следующим образом: $X^* = C^*$, $X = C$, $\mathbf{G} = \mathbf{S}$, $V = L\mathbb{1}_K$, (11b) состоит из (8с), (11с) состоит из (8d) и (8е).

Таким образом, необходимо объяснить, как находить начальную точку (п. 1 алгоритма 3), решение подзадачи квадратичного программирования и множители Лагранжа (п. 4 алгоритма 3) применительно к частному случаю задачи (8b).

Обозначим вкратце те особенности, которые помогают получить быстрое решение.

Для выполнения пункта 4 алгоритма 3 требуется уметь решать задачу (12а) с ограничениями (12b). Положив $\mathbf{A} = [A_i : i \in W_k]$, $\mathbf{A} \in \mathbb{R}^{K \times m}$, ограничения (12b) можно записать как $\mathbf{A}^T P_k = 0$, где m — количество активных ограничений.

Для решения поставленной вспомогательной задачи (12а) есть явная формула обобщенного метода наименьших квадратов:

$$P_k^* = \overline{\mathbf{A}} \left(\overline{\mathbf{A}}^T \mathbf{G} \overline{\mathbf{A}} \right)^{-1} \overline{\mathbf{A}}^T G_k,$$

где матрица $\overline{\mathbf{A}} \in \mathbb{R}^{K \times (K-m)}$ состоит из столбцов, составляющих базис ортогонального дополнения к базису столбцов матрицы \mathbf{A} . Обычно m велико, поэтому $(K-m)$ мало, что позволяет быстро искать решение подзадачи.

В случае задачи 3 матрица \mathbf{G} имеет простой вид, и ее можно умножить на вектор за время $O(K)$. Матрица \mathbf{A} — разреженная, в ней содержится максимум 2 ненулевых коэффициента в каждом столбце. За счет этого матрицу $\overline{\mathbf{A}}$ можно также быстро найти и хранить, используя $O(K)$ памяти. Из-за быстрого умножения матрицы \mathbf{A} для вычисления формулы обобщенного метода наименьших квадратов удобно и эффективно использовать метод сопряженных градиентов, вычислительная сложность которого равна $O(K(K-m))$.

Аналогично за счет разреженности матрицы \mathbf{A} можно быстро находить множители Лагранжа за время $O(K)$. Таким образом, вычислительная сложность одного шага алгоритма 3 при решении подзадачи (8b) равна $O(K(K-m))$.

Есть две стратегии выбора начальной точки: первая применяется при малом размере задачи и заключается в том, чтобы назначить первым t точкам, где t перебирается по целочисленной сетке до $\lfloor K/2 \rfloor$, максимальный вес (то есть (8е)), а оставшимся — минимальный (то есть (8d)), после чего выбрать то t , где достигается наименьшее значение целевой функции $f(C)$.

Вторая стратегия используется при $j = 1$ и основана на следующем наблюдении: решения задач при примерно одинаковом отношении N к L и одинаковом α схожи. Например, на рис. 1, a, b изображены два отнормированных (умноженных на N) решения задачи 2 при $N = 200$, $L = 60$, $\alpha = 0.1$ и $N = 1000$, $L = 300$, $\alpha = 0.1$ соответственно.

Схема эвристики следующая: зафиксируем параметр масштаба $0 < \gamma < 1$ (на практике хорошими значениями γ являются 0.5–0.7), найдем решение задачи при

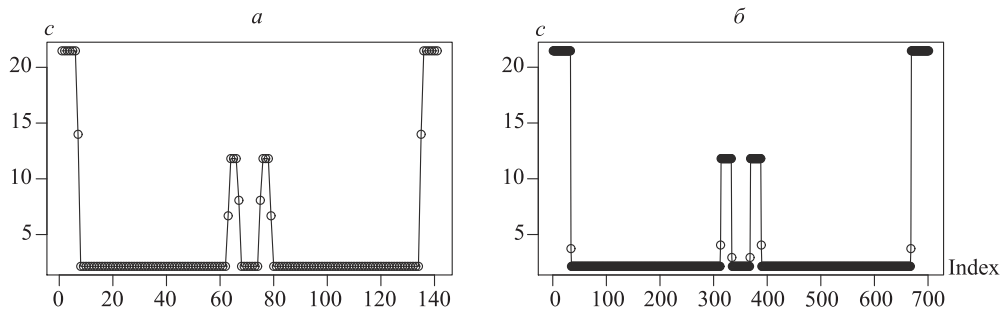


Рис. 1. Два решения задачи 2 при одинаковом отношении N к L .

$N_\gamma \approx \gamma N$, $L_\gamma \approx \gamma L$, $\alpha_\gamma = \alpha$, после чего используем решение задачи меньшего порядка для выбора начального рабочего множества и начальной точки. Формально, алгоритм выглядит следующим образом.

Алгоритм 4. Вход: параметр γ .

Результат: Начальное рабочее множество ограничений W_0 и стартовая точка X_0 .

1. Положить $K_\gamma = \lceil \gamma K \rceil$, $L_\gamma = \lceil \gamma L \rceil$, $N_\gamma = L_\gamma + K_\gamma - 1$.
2. Определить функцию $s(i) = \lceil \frac{(i-1)(K_\gamma-1)}{K_\gamma-1} + 1 \rceil$ перехода к уменьшенной размерности.
3. Найти масштабированное решение $C_\gamma = (c_{\gamma,1}, \dots, c_{\gamma,K_\gamma})^T$ задачи (8b) при $K = K_\gamma$, $L = L_\gamma$, $\alpha = \alpha$, $j = j_\gamma = 1$.
4. Добавить к рабочему множеству W индексы всех ограничений вида (8с).
5. Для $i = 2, 3, \dots, \lceil K/2 \rceil$
 - 1) вычислить $i_\gamma = s(i)$;
 - 2) если $c_{\gamma,i_\gamma} - \alpha c_{\gamma,j_\gamma} = 0$, добавить в W индекс ограничения вида $\{c_i - \alpha c_j \geq 0\}$ из (8d) и перейти к следующей итерации;
 - 3) если $c_{\gamma,i_\gamma} = c_{\gamma,j_\gamma}$, добавить в W индекс ограничения вида $\{c_j - c_i \geq 0\}$ из (8e) и перейти к следующей итерации.
6. Для $k = 1, 2, \dots$
 - 1) решить следующую задачу квадратичного программирования, вычислить значение точки минимума C^* :

$$C^* = \arg \min_{C \in C_k} f(C), \quad C_k = \{C \mid A_i^T C = 0, \quad i \in W\},$$

где $f(C)$ определено в (6а);

- 2) проверить все условия вида (8d) и (8e) для точки C^* ; если все они выполняются, ответ $X_0 = C^*$, $W_0 = W$ и STOP, иначе добавить к рабочему множеству W индекс любого ограничения, которое не выполняется.

Заметим, что шаг 6 алгоритма 4, необходимый для того, чтобы получить корректную начальную точку, совершит не более $\lceil K/2 \rceil$ итераций вследствие того, что каждая итерация уменьшает размерность подпространства C_k , в которой лежит начальная точка C^* , на единицу.

Таким образом, при использовании данной эвристики получаем рекурсивный алгоритм, который уменьшает размерность задачи в γ раз до тех пор, пока K не станет достаточно малым для использования первой стратегии.

6. Численный эксперимент. Ниже приведем численное сравнение эффективности полученных с помощью алгоритма 1 весов в задаче оценки сигнала рядом конечного ранга методом Oblique Cadzow [6]. Сравнение было проведено на примере синусоидального сигнала.

Был взят сигнал $S = (s_1, \dots, s_N)$ длины $N = 40$ и ранга $r = 2$, имеющий вид

$$s_k = 5 \sin \frac{2\pi k}{6}, \quad k = 1, \dots, N.$$

Рассматривался ряд вида $X = S + N$, где N — гауссовский белый шум с нулевым средним и единичной дисперсией. Точность оценки сигнала \hat{S} измерялась с помощью корня из среднего по точкам ряда и по 1000 реализациям ряда квадрата отклонения от сигнала S . Эту меру будем называть RMSE (*root mean-square error*) оценки сигнала. В качестве меры скорости сходимости алгоритма Oblique Cadzow используется среднее число итераций до остановки. Длина окна зафиксирована равной $L = 20$. Был использован следующий критерий остановки метода Oblique Cadzow: $\|\mathcal{T}_L^{-1}(\mathbf{Y}_k) - \mathcal{T}_L^{-1}(\mathbf{Y}_{k+1})\|^2/N < 10^{-8}$. Рассматривались веса, полученные алгоритмом 1 (тип I) и прежним методом Cadzow(α) [6, предл. 5] (тип II). Одинаково отнормированные веса в пространстве рядов при $\alpha = 2^{-2}$ и $\alpha = 2^{-4}$ изображены на рис. 2, а, б соответственно.

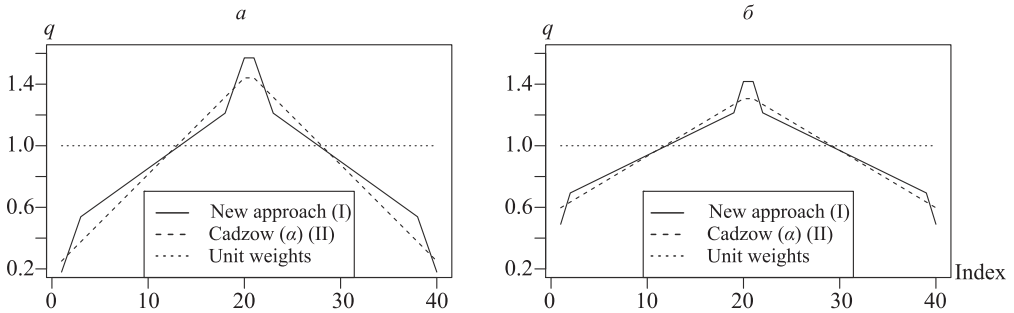


Рис. 2. Полученные веса в пространстве рядов при различных α : $\alpha = 2^{-2}$ (а), $\alpha = 2^{-4}$ (б).

Результаты сравнения приведены в таблице. Заметим, что результаты при $\alpha = 1$ и $\alpha = 2^{-5}, 2^{-6}$ совпадают, так как численно равны матричные веса, полученные двумя различными методами. В целом методы дают сравнимые результаты, а при $\alpha = 0.5$ и $\alpha = 0.25$ новый подход дает оценку лучше, чем метод Cadzow(α). Также заметим, что заявленные соотношения между α и точностью оценивания, а также α и скоростью сходимости метода Oblique Cadzow, выполняются и для нового метода. При малых α точность оценки существенно лучше, чем при $\alpha = 1$, при этом точность убывает с ростом α .

**Сравнение методов по RMSE и среднему числу итераций
при различных α и алгоритмах получения весов (I, II)**

α	RMSE, I	RMSE, II	Iterations, I	Iterations, II
1	0.3811	0.3811	8.44	8.44
2^{-1}	0.3615	0.3682	8.67	8.71
2^{-2}	0.3475	0.3529	9.76	9.55
2^{-3}	0.3371	0.3388	11.97	11.61
2^{-4}	0.3300	0.3293	14.73	15.95
2^{-5}	0.3247	0.3247	24.37	24.37
2^{-6}	0.3231	0.3231	39.95	39.95

Заключение. Для решения задачи поиска весов было применено несколько нетривиальных переходов. Сформулированная в терминах минимизации квадратичного функционала задача 2 содержит K^2 линейных ограничений, что не позволяет эффективно искать решение на практике. Поэтому была выведена эквивалентная задача 3, решение которой — минимум из решений $K/2$ задач квадратичного программирования с количеством ограничений порядка K . Преимущество формулировки 3 состоит в том, что перебор задач КП можно остановить заранее, используя разработанный метод остановки, реализованный в алгоритме 2. На практике перебор ограничивается уже на первой итерации, вследствие чего решение находится эффективно.

Качество полученных алгоритмом 1 весов было проверено численным экспериментом. Для длины ряда N , кратной длине окна L , результаты сравнимы с прежним алгоритмом Cadzow(α), но новый алгоритм позволяет находить решение для произвольных N и L .

Литература

1. Golyandina N., Nekrutkin V., Zhigljavsky A. Analysis of Time Series Structure: SSA and Related Techniques. Chapman & Hall/CRC, 2001.
2. Broomhead D., King G. Extracting qualitative dynamics from experimental data // Physica D. 1986. Vol. 20. P. 217–236.
3. Vautard R., Yiou P., Ghil M. Singular-Spectrum Analysis: a Toolkit for short, noisy chaotic signals // Physica D. 1992. Vol. 58. P. 95–126.
4. Elsner J. B., Tsonis A. A. Singular Spectrum Analysis: a New Tool in Time Series Analysis. Plenum, 1996.
5. Stoica P., Moses R. L. Spectral analysis of signals. Pearson Prentice Hall Upper Saddle River, New Jersey, 2005. Vol. 452.
6. Zvonarev N., Golyandina N. Iterative algorithms for weighted and unweighted finite-rank time-series approximations // Statistics and Its Interface. 2017. Vol. 10. P. 5–18.
7. Gillard J., Zhigljavsky A. A. Stochastic algorithms for solving structured low-rank matrix approximation problems // Communication in Nonlinear Science and Numerical Simulation. 2015. Vol. 21, N 1. P. 70–88.
8. Zhigljavsky A., Golyandina N., Gryaznov S. Deconvolution of a discrete uniform distribution // Statistics & Probability Letters. 2016. Vol. 118. P. 37–44.
9. Габурич М. К., Малоземов В. Н. Экстремальные задачи с линейными ограничениями. Учебное пособие. Л.: Изд-во Ленингр. ун-та, 1984.
10. Nocedal J., Wright S. Numerical optimization. Springer Science & Business Media, 2006.

Статья поступила в редакцию 11 марта 2016 г.

Сведения об авторе

Звонарев Никита Константинович — аспирант; nikitazvonarev@gmail.com

SEARCH OF WEIGHTS IN PROBLEM OF WEIGHTED FINITE-RANK TIME-SERIES APPROXIMATION

Nikita K. Zvonarev

St. Petersburg State University, Universitetskaya nab., 7–9, St. Petersburg, 199034, Russian Federation; nikitazvonarev@gmail.com

The problem of weighted finite-rank time-series approximation is considered for signal estimation. Finding of weights, which lead to improvement of estimate accuracy, is examined. An effective method for numerical search of weights is constructed and proved. The theory of quadratic optimization is used. For effective algorithm, the problem of quadratic optimization with large number of linear constraints reduced to a sequence of problems with smaller number constraints, which is completed by a stopping criterion. For the algorithm deduction, equivalence of different statements of the original optimization problem is proved. A numerical simulation is performed to approve the algorithm efficiency and improvement of the estimate accuracy. Refs 10. Figs 2. Table 1.

Keywords: time series, time series of finite rank, Cadzow's iterative method, weighted low-rank approximation, oblique singular value decomposition, quadratic programming.

References

1. Golyandina N., Nekrutkin V., Zhigljavsky A., *Analysis of Time Series Structure: SSA and Related Techniques* (Chapman & Hall/CRC, 2001).
2. Broomhead D., King G., "Extracting qualitative dynamics from experimental data", *Physica D.* **20**, 217–236 (1986).
3. Vautard R., Yiou P., Ghil M., "Singular-Spectrum Analysis: a Toolkit for short, noisy chaotic signals", *Physica D.* **58**, 95–126 (1992).
4. Elsner J.B., Tsonis A.A., *Singular Spectrum Analysis: a New Tool in Time Series Analysis* (Plenum, 1996).
5. Stoica P., Moses R. L., *Spectral analysis of signals* **452** (Pearson Prentice Hall Upper Saddle River, New Jersey, 2005).
6. Zvonarev N., Golyandina N., "Iterative algorithms for weighted and unweighted finite-rank time-series approximations", *Statistics and Its Interface* **10**, 5–18 (2017).
7. Gillard J., Zhigljavsky A. A., "Stochastic algorithms for solving structured low-rank matrix approximation problems", *Communication in Nonlinear Science and Numerical Simulation* **21**(1), 70–88 (2015).
8. Zhigljavsky A., Golyandina N., Gryaznov S., "Deconvolution of a discrete uniform distribution", *Statistics & Probability Letters* **118**, 37–44 (2016).
9. Gavurin M. K., Malozemov V. N., *Extremal problems with linear constraints* (Izdat. Leningrad State Univ., Leningrad, 1984) [in Russian].
10. Nocedal J., Wright S., *Numerical optimization* (Springer Science & Business Media, 2006).

Для цитирования: Звонарев Н.К. Поиск весов в задаче взвешенной аппроксимации временным рядом конечного ранга // Вестник Санкт-Петербургского университета. Серия 1. Математика. Механика. Астрономия. 2016. Т. 3 (61). Вып. 4. С. 570–581. DOI: 10.21638/11701/spbu01.2016.406

For citation: Zvonarev N.K. Search of weights in problem of weighted finite-rank time-series approximation. *Vestnik of Saint Petersburg University. Series 1. Mathematics. Mechanics. Astronomy*, 2016, vol. 3 (61), issue 4, pp. 570–581. DOI: 10.21638/11701/spbu01.2016.406