Санкт-Петербургский государственный университет

Прикладная математика и информатика

Статистическое моделирование

Третьякова Александра Леонидовна

Робастные варианты метода анализа сингулярного спектра

Научно-исследовательская работа

Научный руководитель:

к. ф.-м. н., доцент Н. Э. Голяндина

Санкт-Петербург

2019

Оглавление

Введение

В реальной жизни часто возникают задачи исследования различных процессов с течением времени. Пусть имеется $x(t)$ — функция, описывающая некоторый процесс во времени. Если произвести измерения через одинаковые промежутки времени t_i, где $i = 1, \ldots, N$, тогда $x_i = x(t_i)$ представляют собой временной ряд $\mathsf{X} = (x_1, \ldots, x_N)$.

Для решения многих задач, к примеру, экономических, таких как планирование производства или инвестиций, оказывается полезным на основе данных за предшествующий период выделить основную динамику и тенденции, а также спрогнозировать развитие процесса. Работа посвящена изучению одного из методов исследования временных рядов — «Гусеница»-SSA (Singular Spectrum Analysis). Метод нашел свое применение в задачах исследования климатических явлений [1], динамических систем [2, 3] и во многих других областях. Данный метод позволяет получить разложение интересующего нас временного ряда $\mathsf{X} = (x_1, \ldots, x_N)$ на интерпретируемые аддитивные составляющие:

$$\mathsf{X} = \mathsf{S} + \mathsf{R},$$

где S — сигнал, R — шум, например, некоторый стационарный процесс.

Традиционно при постановке задачи отделения сигнала от шума шум предполагается гауссовским. Однако на практике часто возникают выделяющиеся наблюдения или выбросы, которые можно интерпретировать как ошибки в данных или сбои измерительного прибора, значительно большие, чем размер шума. Отфильтровать их оказывается непростой задачей, необходимо сначала разобраться со структурой ряда, чтобы понять, что данное значение является выбросом. Поэтому разработка исходно устойчивых к выбросам методов представляет интерес.

Ранее в работе [4] уже были предложены несколько устойчивых к выбросам вариантов метода, но они оказались слишком трудоемкими и алгоритмы работали очень долго. Поэтому требуются модификации этих методов, которые бы оставались устойчивыми, но время работы алгоритмов было меньше. В данной работе стоит задача предложить менее трудоемкие модификации робастных методов и сравнить их с базовым методом.

В методе SSA при выделении сигнала используются два проектора, которые могут строиться по различным нормам. Один из проекторов — это проектор на пространство ганкелевых матриц, второй — проектор на множество матриц ранга, не превосходящего r. В стандартном методе SSA используются проекторы в пространстве матриц по норме

$\mathbb{L}_2$ (норма Фробениуса).

В качестве модификаций в работе [4] рассматривался стандартный прием использования аппроксимации (проекции) по норме в $\mathbb{L}_1$ вместо $\mathbb{L}_2$. Если построение проектора на ганкелевы матрицы по норме $\mathbb{L}_1$ не представляет трудности, то вычисление проектора на матрицы ранга, не превосходящего r, по норме $\mathbb{L}_1$ не имеет решения в замкнутой форме. Имеются методы, численно решающие приближенные задачи, но не известно достаточно хороших методов для задачи, которую требуется решить при построении проектора на матрицы ранга, не превосходящего r.

Введем классификацию методов согласно используемым нормам. В общем случае методы будут называться LiSVD-LjH-SSA, где i, j могут быть равны 1 или 2. LiSVD означает проектор на матрицы ранга, не превосходящего r, по норме в пространстве $\mathbb{L}_2$ (L2SVD) или $\mathbb{L}_1$ (L1SVD), а LjH — проектор на пространство ганкелевых матриц по норме $\mathbb{L}_2$ (L2H) или $\mathbb{L}_1$ (L1H). Для более короткой записи будем называть стандартный метод с двумя проекторами по норме в $\mathbb{L}_2$ методом L2-SSA, а метод с двумя проекторами в $\mathbb{L}_1$ — L1-SSA.

Структура работы следующая. В главе 1 опишем базовый алгоритм метода SSA, введем необходимые понятия и обозначения, обсудим выбор параметров метода на основе теории метода SSA.

В главе 2 рассматривается общая схема методов без указания конкретной нормы. Один из ключевых вопросов – это способы нахождения $\mathbb{L}_1$-проектора на множество матриц ранга, не превосходящего r. В работе рассматриваются два метода проектирования, один из них взят из статьи [5], он использует метод взвешенной медианы для решения задачи минимизации. Второй метод рассматривается в R-пакете в рамках построения устойчивого к выбросам анализа главных компонент [6]. Приведены алгоритмы для каждого из способов. Также проведено теоретическое сравнение вариантов метода L1-SSA между собой.

Еще одной идеей для достижения устойчивости метода к выделяющимся наблюдениям является присвоение значениям в точках, содержащих выбросы, меньший вес. Был рассмотрен алгоритм, описанный в статье [7], использующий взвешенный метод наименьших квадратов. Данный метод оказался неподходящим для нестационарных рядов с растущей или убывающей амплитудой. Поэтому была предложена модификация метода. Она также описана в главе 2.

В конце главы 2 произведен подсчет и сравнение теоретической трудоемкости описанных методов. Было рассмотрено два случая: когда траекторная матрица близка к квадратной, и случай с вытянутой траекторной матрицей. В обоих случаях теоретическая трудоемкость последовательного метода из статьи [6] оказывается меньше теоретической трудоемкости метода из статьи [5].

Глава 3 содержит численные сравнения, в которых исследуется влияние выброса на результат восстановления сигнала. В данной главе описывается сравнение четырех вариантов метода L1-SSA между собой и с классическим методом L2-SSA. Сравнение проводилось при отсутствии выделяющихся наблюдений, при 1% выбросов в случайных точках ряда и при 5% выбросов.

Сравнение проводилось на двух примерах. Один из них уже был представлен в работе [4], но было добавлено большее количество выбросов. Было показано, что метод, использующий взвешенный метод наименьших квадратов оказывается наиболее устойчивым среди других методов для первого примера. Предложенная в пункте 2.4 модификация для второго ряда дает наименьшую ошибку в присутствии выделяющихся наблюдений.

В заключении описаны основные результаты работы, подведены итоги.

Работа в текущем семестре заключалась в рассмотрении идеи с добавлением маленьких весов точкам, содержащим выбросы. Был рассмотрен метод, использующий взвешенный метод наименьших квадратов. Так как были обнаружены недостатки этого метода в случае рядов с растущей или убывающей амплитудой, то была разработана модификация с целью борьбы с этими недостатками. Проведено сравнение теоретических трудоемкостей рассматриваемых методов и их сравнение между собой. На трех примерах было проведено сравнение методов, а также проверена значимость сравнений.

Глава 1

Стандартный метод SSA и его свойства

1.1. Алгоритм метода SSA

Кратко опишем базовый алгоритм метода «Гусеница»-SSA, следуя [8].

1.1.1. Вложение

На первом шаге алгоритма выбирается некоторое целое число L: $1 < L < N$, называемое *длиной окна*. Исходный временной ряд переводится в последовательность многомерных векторов длины L. В результате образуются $K = N - L + 1$ векторов вложения

$$X_i = (x_i, \ldots, x_{i+L-1})^{\mathrm{T}}, 1 \le i \le K.$$

Траекторной матрицей ряда X называется матрица

$$\mathbf{X} = [X_1 : \ldots : X_K] = \begin{pmatrix} x_1 & x_2 & x_3 & \ldots & x_K \\ x_2 & x_3 & x_4 & \ldots & x_{K+1} \\ x_3 & x_4 & x_5 & \ldots & x_{K+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_L & x_{L+1} & x_{L+2} & \ldots & x_N \end{pmatrix}.$$

Заметим, что построенная таким образом траекторная матрица $\mathbf{X}$ является *ганкелевой*, т.е. элементы, находящиеся на диагоналях $i + j = \text{const}$, равны между собой.

1.1.2. Сингулярное разложение

Пусть $\mathbf{S} = \mathbf{X}\mathbf{X}^{\mathrm{T}}$, обозначим $\lambda_1 \ge \lambda_2 \ge \ldots \ge \lambda_d > 0$ — ненулевые собственные числа матрицы $\mathbf{S}$, $U_1, \ldots, U_d$ — ортонормированная система собственных векторов матрицы $\mathbf{S}$, соответствующих собственным числам. *Сингулярным разложением* матрицы $\mathbf{X}$ называется разложение

$$\mathbf{X} = \mathbf{X}_1 + \ldots + \mathbf{X}_d = \sum_{i=1}^{d} \sqrt{\lambda_i} U_i V_i^{\mathrm{T}},$$

где $\sqrt{\lambda_i}$ — *сингулярные числа*, U_i — *левые сингулярные вектора*, $V_i = \frac{1}{\sqrt{\lambda_i}} X^{\mathrm{T}} U_i$ — *правые сингулярные вектора*.

Набор $(\sqrt{\lambda_i}, U_i, V_i)$ назовем *i-ой собственной тройкой* сингулярного разложения.

1.1.3. Группировка

Разделим множество индексов $\{1, \ldots, d\}$ на m дизъюнктных подмножеств $I_1, \ldots, I_m$. Пусть $I = \{i_1, \ldots, i_p\}$. Тогда *результирующая матрица* $\mathbf{X}_I$ имеет вид

$$\mathbf{X}_I = \mathbf{X}_{i_1} + \ldots + \mathbf{X}_{i_p}.$$

Таким образом, получаем разложение матрицы $\mathbf{X}$ в сгруппированном виде

$$\mathbf{X} = \mathbf{X}_{I_1} + \ldots + \mathbf{X}_{I_m}.$$

1.1.4. Диагональное усреднение

На последнем шаге каждая матрица $\mathbf{X}_{I_i}$ переводится в новый ряд с помощью усреднения элементов матрицы вдоль антидиагоналей $i + j = k + 1$. Применяя диагональное усреднение к результирующим матрицам, получаем ряды $\tilde{\mathsf{X}}^{(k)} = (\tilde{x}_1^k, \ldots, \tilde{x}_N^k)$.

В результате получаем разложение исходного ряда $(x_1, \ldots, x_N)$ в сумму m рядов:

$$x_n = \sum_{k=1}^{m} \tilde{x}_n^{(k)}.$$

1.2. Разделимость

Понятие разделимости подробно описано в [8]. Однако условия разделимости являются слишком жесткими и редко выполнены в реальных задачах. Поэтому введем понятие приближенной разделимости.

Для ряда $\mathsf{X} = (x_1, \ldots, x_N)$ положим $X_{i,j} = (x_i, \ldots, x_j)$, $1 \le i \le j < N$. Пусть $\mathsf{X}_N^{(1)} = (x_1^{(1)}, \ldots, x_N^{(1)})$, $\mathsf{X}_N^{(2)} = (x_1^{(2)}, \ldots, x_N^{(2)})$. Пусть

$$\rho_{i,j}^{(M)} = \frac{(X_{i,i+M-1}^{(1)}, X_{j,j+M-1}^{(2)})}{\|X_{i,i+M-1}^{(1)}\| \, \|X_{j,j+M-1}^{(2)}\|}, \quad i,j \ge 1, \; M \le N - \max(i,j),$$

где $\| \, . \, \|$ — евклидова норма, $(\, . \, , \, . \,)$ — скалярное произведение векторов. Если знаменатель равен нулю, то предполагаем, что $\rho_{i,j}^{(M)} = 0$.

Число $\rho_{i,j}^{(M)}$ равно косинусу угла между векторами $X_{i,i+M-1}^{(1)}$ и $X_{j,j+M-1}^{(2)}$.

Определение 1. *Ряды* $\mathsf{X}_N^{(1)}$ *и* $\mathsf{X}_N^{(2)}$ *называются* ε*-разделимыми при длине окна* L*, если*

$$\rho^{(L,K)} = \max(\max_{1 \le i,j \le K} |\rho_{i,j}^{(L)}|, \max_{1 \le i,j \le L} |\rho_{i,j}^{(K)}|) \le \varepsilon,$$

$K = N - L + 1$. *Если число* ε *мало, то ряды называются приближенно разделимыми.*

Если $\rho^{(L,K)} = 0$, то это соответствует точной разделимости.

Введем понятие асимптотической разделимости. Пусть $\mathsf{X}^{(1)} = (x_1^{(1)}, \ldots, x_n^{(1)}, \ldots)$, $\mathsf{X}^{(2)} = (x_1^{(2)}, \ldots, x_n^{(2)}, \ldots)$ и для любого $N > 2$ ряды $\mathsf{X}_N^{(1)}$ и $\mathsf{X}_N^{(2)}$ состоят из первых N членов рядов $\mathsf{X}^{(1)}$ и $\mathsf{X}^{(2)}$. Тогда если выбрать последовательность длин окон $1 < L = L(N) < N$, получим последовательность $\rho_N = \rho^{(L(N),K(N))}$.

Определение 2. *Если $\rho^{(L(N),K(N))} \to 0$ при некоторой последовательности $L = L(N)$, $N \to \infty$, то ряды $\mathsf{X}^{(1)}$ и $\mathsf{X}^{(2)}$ называются асимптотически $L(N)$-разделимыми. Если для любой последовательности $L(N)$: $L(N) \to \infty, K(N) \to \infty$ ряды $\mathsf{X}^{(1)}$ и $\mathsf{X}^{(2)}$ асимптотически $L(N)$-разделимы, то они называются асимптотически разделимыми.*

Асимптотическая раделимость выполняется для более широкого класса рядов, чем точная разделимость. К примеру, $e^{\alpha n}$ и $\sin(2\pi\omega n)$, где $\alpha \neq 0,\ \omega \in (0, 0.5]$, асимптотически разделимы.

Для достижения лучшей разделимости необходимо выбирать большую длину окна ($L \sim N/2$). Большая длина окна позволяет выделить сигнал из зашумленного ряда, отделить тренд от периодических компонент. Не имеет смысла брать длину окна, большую чем половина длины ряда, а маленькая длина окна может привести к смешиванию компонент ряда.

1.3. Ранг ряда

Пусть $\mathsf{X}_N = \mathsf{X}_N^{(1)} + \mathsf{X}_N^{(2)}$ и ряды $\mathsf{X}_N^{(1)}$ и $\mathsf{X}_N^{(2)}$ разделимы. Тогда в сингулярном разложении ряда X_N часть слагаемых относится к сингулярному разложению ряда $\mathsf{X}_N^{(1)}$, а другая часть — к сингулярному разложению ряда $\mathsf{X}_N^{(2)}$. Необходимо выяснить, сколько слагаемых относится к первому ряду и как их идентифицировать.

Рассмотрим ряд $\mathsf{X}_N = (x_1, \ldots, x_N)$, пусть L – длина окна.

Обозначим $\mathcal{L}^{(L)} = \mathrm{span}(X_1, \ldots, X_K)$ — траекторное пространство ряда X_N, где $X_i = (x_i, \ldots, x_{i+L-1})^T$ — векторы вложения, $1 \leq i \leq K$.

Определение 3. *Пусть $0 < d \leq \min(L, K)$. Будем говорить, что ряд X_N имеет L-ранг d, если $\dim \mathcal{L}^{(L)} = d$.*

Например, в случае экспоненциального ряда $e^{\alpha n}$ для любых N и L ранг ряда равен 1, а ранг гармонического ряда $\sin(2\pi\omega n + \phi)$ равен 2 при $\omega < 1/2$ и 1 при $\omega = 1/2$, $\phi \in [0, 2\pi)$.

Глава 2

Модификации метода SSA с проекторами по некоторой норме

2.1. Схема методов

Пусть имеется временной ряд $\mathsf{X} = (x_1, \ldots, x_N)$.

Выбирается длина окна L, и исходный временной ряд переводится в последовательность многомерных векторов длины L. В результате образуются $K = N - L + 1$ векторов вложения

$$X_i = (x_i, \ldots, x_{i+L-1})^{\mathrm{T}}, 1 \leq i \leq K.$$

Обозначим $\mathcal{M}$ — пространство матриц $L \times K$,

$\mathcal{M}_{\mathcal{H}}$ — пространство ганкелевых матриц $L \times K$,

$\mathcal{M}_r$ — пространство матриц ранга, не превосходящего r.

Определим следующие операторы:

- Оператор вложения $\mathcal{T} : \mathbb{R}^N \to \mathcal{M}_{\mathcal{H}} : \mathcal{T}(\mathsf{X}) = \mathbf{X}$.
- $\Pi_r : \mathcal{M} \to \mathcal{M}_r$ — проектор на множество матриц ранга, не превосходящего r, по некоторой норме в пространстве матриц.
- $\Pi_{\mathcal{H}} : \mathcal{M} \to \mathcal{M}_{\mathcal{H}}$ — проектор на пространство ганкелевых матриц по некоторой норме в пространстве матриц.

В результате применения данных операторов получаем оценку сигнала:

$$\tilde{\mathsf{S}} = \mathcal{T}^{-1}\Pi_{\mathcal{H}}\Pi_r\mathcal{T}(\mathsf{X}).$$

Это соответствует алгоритму SSA, описанному в разделе 1.1, для случая, когда восстановление производится по одной группе, состоящей из первых r компонент.

Проекторы Π_r и $\Pi_{\mathcal{H}}$ можно строить по различным нормам. С точки зрения вычислений, удобно выбирать $\mathbb{L}_2$-норму для построения проекторов на пространство ганкелевых матриц и матриц ранга, не превосходящего r, поскольку целевая функция является гладкой и выпуклой, и решить задачу минимизации довольно просто, можно

даже говорить о задании решения в явной форме. Однако при наличии выбросов норма Фробениуса оказывается недостаточно устойчивой. Норма в пространстве $\mathbb{L}_1$ является более устойчивой к выделяющимся наблюдениям, однако сложность в ее использовании состоит в негладкой и невыпуклой строго целевой функции, поэтому возникает проблема в применении стандартных методов оптимизации.

В работе будут рассмотрены проекторы по нормам в пространствах $\mathbb{L}_2$ и $\mathbb{L}_1$. В стандартном методе SSA оба проектора Π_r и $\Pi_{\mathcal{H}}$ строятся по норме в пространстве $\mathbb{L}_2$. Будем называть его L2-SSA. Метод с проекцией на множество ганкелевых матриц в $\mathbb{L}_1$ и проекцией на множество матриц ранга, не превосходящего r, в $\mathbb{L}_2$ назовем L2SVD-L1H-SSA. Метод с обеими проекциями в $\mathbb{L}_1$ будем называть L1-SSA.

2.2. Способы нахождения $\mathbb{L}_1$-проектора на множество матриц ранга, не превосходящего r

В отличие от проектора на множество матриц ранга r по норме Фробениуса, построение данного проектора в пространстве $\mathbb{L}_1$ является вычислительно сложной задачей.

Рассмотрим несколько методов построения проектора на множество матриц ранга, не превосходящего r, по норме в пространстве $\mathbb{L}_1$.

2.2.1. Метод, использующий взвешенную медиану

Рассмотрим метод вычисления проекции на множество матриц ранга, не превосходящего r, по норме в пространстве $\mathbb{L}_1$, описанный в статье [5].

Для начала введем обозначения. Пусть

$$\mathbf{Y} = [Y_1 : \ldots : Y_p] = \begin{pmatrix} y_{11} & \ldots & y_{1p} \\ \vdots & \ddots & \vdots \\ y_{n1} & \ldots & y_{np} \end{pmatrix}.$$

В наших обозначениях n соответствует длине окна L, p соответствует $K = N - L + 1$, где N — длина ряда, r — ранг.

Опишем подробно каждый шаг алгоритма.

1. Инициализация. Пусть

$$m = \begin{pmatrix} \operatorname{med}(|y_{11}|, \ldots, |y_{1p}|) \\ \vdots \\ \operatorname{med}(|y_{n1}|, \ldots, |y_{np}|) \end{pmatrix}.$$

Возьмем в качестве начального значения $U_1^0 = m/\|m\|_2$.

2. Находим проекцию в $\mathbb{L}_1$ каждого столбца матрицы $\mathbf{Y}$ на вектор $U_1^{(k)}$, то есть для каждого $j = 1, \ldots, p$ решаем задачу минимизации

$$\|(y_{1j}, \ldots, y_{nj})^{\mathrm{T}} - c_j * (u_1, \ldots, u_n)^{\mathrm{T}}\|_1 \to \min_{c_j}$$

методом взвешенной медианы (обозначим здесь $U_1^{(k)} = (u_1, \ldots, u_n)^{\mathrm{T}}$).

Далее нормируем полученный вектор $C = (c_1, \ldots, c_p)$ и полагаем

$$V_1^{(k)} = C/\|C\|_2.$$

3. Находим проекцию в $\mathbb{L}_1$ каждой строки матрицы $\mathbf{Y}$ на вектор $V_1^{(k)}$, то есть для каждого $i = 1, \ldots, n$ решаем задачу

$$\|(y_{i1}, \ldots, y_{ip}) - d_i * (v_1, \ldots, v_p)\|_1 \to \min_{d_i}$$

методом взвешенной медианы.

Далее нормируем полученный вектор $D = (d_1, \ldots, d_n)^{\mathrm{T}}$ и полагаем

$$U_1^{(k+1)} = D/\|D\|_2.$$

4. Критерий остановки: продолжаем выполнять шаги 2 и 3, пока изменение в $\mathbb{L}_2$ вектора U_1 превосходит ε, то есть

$$\text{While} \quad \sum_{i=1}^{n} (U_{1_i}^{(k+1)} - U_{1_i}^{(k)})^2 > \varepsilon.$$

По умолчанию $\varepsilon = 10^{-10}$.

5. В результате имеем U_1^*, V_1^*. Далее находим λ_1^*, решая задачу

$$\sum_{i=1}^{n} \sum_{j=1}^{p} |y_{ij} - \lambda_1 U_1^* V_1^{*\mathrm{T}}| \to \min_{\lambda_1}.$$

6. Из матрицы $\mathbf{Y}$ вычитаем первую компоненту

$$\mathbf{Y} = \mathbf{Y} - \lambda_1^* \mathbf{U}_1^* \mathbf{V}_1^{*\mathrm{T}}$$

и продолжаем искать остальные собственные тройки.

В итоге получаем представление матрицы $\mathbf{Y}$ в виде $\mathbf{U}\mathbf{\Lambda}\mathbf{V}^{\mathrm{T}}$, где $\mathbf{U}$ составлена из $U_1, \ldots, U_p$, а $\mathbf{V}$ состоит из $V_1, \ldots, V_p$. Точнее, мы нашли решение задачи минимизации функции $\|\mathbf{Y} - \mathbf{U}\mathbf{\Lambda}\mathbf{V}^{\mathrm{T}}\|_1$.

Важно отметить, что

- собственные числа на выходе не отсортированы по убыванию,
- собственные вектора получаются не ортогональными, в отличие от L2-SVD.

В пакете pcaMethods [9] имеется метод robustSvd для вычисления проекции в $\mathbb{L}_1$ на множество матриц ранга, не превосходящего r. Более подробно метод описан в статье [10].

2.2.2. Последовательный метод

Стоит задача проектирования матрицы $\mathbf{X}$ на множество матриц ранга, не превосходящего r. Задачу оптимизации можно представить в виде

$$\min_{\mathbf{V},\mathbf{U}} \|\mathbf{X}^{\mathrm{T}} - \mathbf{V}\mathbf{U}^{\mathrm{T}}\|_1 = \sum_{i=1}^{L} \|X_i - \mathbf{V}U_i\|_1,$$

где $\mathbf{V}$ — матрица $K \times r$, $\mathbf{U}$ — матрица $L \times r$. Столбцы матрицы $\mathbf{V}$ определяют главные компоненты. Матрица $\mathbf{E} = \mathbf{U}\mathbf{V}^{\mathrm{T}}$ — проекция $\mathbf{X}$ на множество матриц ранга, не превосходящего r, которую необходимо найти.

В пакете pcaL1 [11] имеется метод l1pca, позволяющий вычислить проекцию в $\mathbb{L}_1$ на множество матриц ранга, не превосходящего r. Подробнее метод описан в статье [6].

Приведем алгоритм в обозначениях предыдущего пункта. Пусть $\mathbf{Y} \in \mathbb{R}^{n \times p}$. Задача выглядит следующим образом:

$$\min_{\mathbf{U},\mathbf{V}} \|\mathbf{Y} - \mathbf{U}\mathbf{V}^{\mathrm{T}}\|.$$

Предположим, что $p \leq n$ и матрица $\mathbf{Y}$ полного ранга.

1. Инициализация $\mathbf{U}(0) \in \mathbb{R}^{n \times p}$, нормировка столбцов $\mathbf{U}(0)$,

2. $t := t + 1$,

3. $\mathbf{V}(t) = \underset{\mathbf{V} \in \mathbb{R}^{p \times p}}{\operatorname{argmin}} \|\mathbf{Y} - \mathbf{U}(t-1)\mathbf{V}^{\mathrm{T}}\|_1$,

4. $\mathbf{U}(t) = \underset{\mathbf{U} \in \mathbb{R}^{n \times p}}{\operatorname{argmin}} \|\mathbf{Y} - \mathbf{U}\mathbf{V}^{\mathrm{T}}(t)\|_1$,

5. Нормировка столбцов $\mathbf{U}(t)$,

6. if $\mathbf{U}(t) \neq \mathbf{U}(t-1)$ (по критерию остановки) then Go to Step 2
   else $\mathbf{U} := \mathbf{U}(t)$; $\mathbf{V} := \mathbf{V}(t)$.

   Критерий остановки:

$$\text{While} \quad ((\max_{i,j} |u_{ij}^{(k)} - u_{ij}^{(k-1)}| > \varepsilon) \text{ и } (\text{iter} \leq \text{MaxIter})),$$

   по умолчанию $\varepsilon = 10^{-4}, \text{MaxIter} = 10$,

7. End.

Решаем задачу, меняя на каждой итерации $\mathbf{U}$ и $\mathbf{V}$ и разбивая исходную задачу на линейные подзадачи. $\mathbf{V}(0)$ можно инициализировать с помощью сингулярного разложения траекторной матрицы $\mathbf{X}$ в пространстве $\mathbb{L}_2$.

2.3. Сравнение методов

В данном разделе сравним методы из разделов 2.2.1 и 2.2.2 между собой.

Метод из раздела 2.2.1 основан на том, чтобы в стандартном методе L2-SSA заменить сингулярное разложение на другое разложение для повышения устойчивости к выбросам. Далее берутся первые r компонент данного разложения, группируются, и применяется диагональное усреднение.

В статье [12], где вводится метод robustSvd, используемый вместо обычного сингулярного разложения в методе из раздела 2.2.1, нет точной формулировки задачи, решаемой данным методом. Также стоит заметить, что при взятии первых r компонент разложения, мы не получим проекцию на пространство матриц ранга, не превосходящего r.

Приведем другие важные отличия между методами.

Важно отметить, что в алгоритме из раздела 2.2.1 поиск решения ведется последовательно для каждой компоненты, то есть по очереди находится каждая компонента,

она вычитается, и далее производится поиск остальных компонент. В методе из раздела 2.2.2 все собственные векторы ищутся параллельно в виде матрицы.

Заметим, что в методе, использующем взвешенную медиану, собственные числа не отсортированы по убыванию. Это может привести к уменьшению точности. Например, если мы нашли неточно первые несколько компонент, вклад которых достаточно мал, а затем ищем компоненту с большим вкладом, то мы уже не так точно найдем эту компоненту.

Еще одно существенное отличие в методах — это то, что задачи минимизации целевой функции решаются по-разному. В одном варианте используется метод взвешенной медианы, а задача в последовательном методе решается с помощью решения задач линейного программирования.

2.4. Взвешенный метод наименьших квадратов

Пусть $\mathbf{Y} = [Y_1, \ldots, Y_p] \in \mathbb{R}^{n \times p}$. Пусть y_i — i-ая строка матрицы $\mathbf{Y}$, $i = 1, \ldots, n$. Напомним, что n — это длина окна L, а p соответствует $K = N - L + 1$, где N — длина ряда, r — ранг.

Идея заключается в замене исходной задачи

$$\min_{\hat{\mathbf{Y}} \in \mathcal{M}_r} \|\mathbf{Y} - \hat{\mathbf{Y}}\|_F.$$

на задачу

$$\min_{\hat{\mathbf{Y}} \in \mathcal{M}_r} \|\mathbf{Y} - \hat{\mathbf{Y}}\|_\rho = \min_{\hat{\mathbf{Y}} \in \mathcal{M}_r} \sum_{i=1}^{n} \sum_{j=1}^{p} \rho(\frac{y_{ij} - \hat{y}_{ij}}{\sigma}). \tag{2.1}$$

Опишем метод, представленный в статье [7], использующий для решения задачи 2.1 взвешенный метод наименьших квадратов с обновлением весов на каждой итерации. Веса должны зависеть от того, насколько большой остаток в точке. В качестве $\rho(x)$ возьмем функцию Тьюки, которая имеет вид

$$\rho(x) = \begin{cases} \frac{1}{6}\alpha^2\{1 - (1 - (\frac{|x|}{\alpha})^2)^3\}, & |x| \le \alpha \\ \frac{1}{6}\alpha^2, & |x| > \alpha \end{cases}, \tag{2.2}$$

где α — параметр. Причина выбора такой функции в качестве метрики заключается в том, что на краях она не так сильно возрастает, как квадратичная функция, а точнее,

данная функция остается постоянной при $|x| > \alpha$, что приводит к более устойчивому к выбросам результату.

Представим матрицу $\hat{\mathbf{Y}}$ в виде $\hat{\mathbf{Y}} = \mathbf{U}\mathbf{V}^{\mathrm{T}}$, где $\mathbf{U} \in \mathbb{R}^{n\times r}, \mathbf{V} \in \mathbb{R}^{p\times r}$. Задача нахождения проекции на множество матриц ранга, не превосходящего r, сводится к задаче

$$\min_{\mathbf{U},\mathbf{V}} \|\mathbf{Y} - \mathbf{U}\mathbf{V}^{\mathrm{T}}\|_{\rho}.$$

Метод имеет параметры α и σ. Их произведение $\alpha\sigma$ по сути является порогом для принятия решения о том, является ли наблюдение выбросом или нет. Опишем алгоритм.

1. Инициализация $\mathbf{U} \in \mathbb{R}^{n\times r}$ и $\mathbf{V} \in \mathbb{R}^{p\times r}$,
2. Выбор параметра α,
3. Вычисление матрицы остатков $\mathbf{R} = \{r_{ij}\}_{i,j=1}^{n,p} = \mathbf{Y} - \mathbf{U}\mathbf{V}^{\mathrm{T}}$,
4. Обновление параметра σ,
5. Вычисление матрицы весов $\mathbf{W}$, используя функцию $w(x) = \frac{\partial\rho(x)}{\partial|x|}\frac{1}{|x|}$:

$$w(x) = \begin{cases} (1 - (\frac{|x|}{\alpha})^2)^2, & |x| \le \alpha \\ 0, & |x| > \alpha \end{cases}, \quad \text{где } x = \frac{1}{\sigma} r_{ij},$$

   то есть для каждого элемента матрицы $\frac{1}{\sigma}\mathbf{R}$ применяем функцию $w(x)$,
6. Вычисление матрицы $\mathbf{U}$ с помощью решения задачи

$$(y_i - \mathbf{V}u_i)^{\mathrm{T}}\mathbf{W}_i(y_i - \mathbf{V}u_i) \to \min_{u_i}, \quad i = 1, \dots n, \tag{2.3}$$

   где $\mathbf{W}_i = \mathrm{diag}(w_i) \in \mathbb{R}^{p\times p}$ — диагональная матрица, составленная из i-ой строки матрицы $\mathbf{W}$.
7. Вычисление матрицы $\mathbf{V}$ с помощью решения задачи

$$(Y_j - \mathbf{U}v_j)^{\mathrm{T}}\mathbf{W}^j(Y_j - \mathbf{U}v^j) \to \min_{v_j}, , \quad j = 1, \dots p, \tag{2.4}$$

   где $\mathbf{W}^j = \mathrm{diag}(W_j) \in \mathbb{R}^{n\times n}$ — диагональная матрица, составленная из j-го столбца матрицы $\mathbf{W}$.
8. Если не выполнен критерий сходимости или максимальное число итераций MaxIterAM не достигнуто, то повторяем шаги 6–7 (alternating minimization),

9. Если не выполнен критерий сходимости или максимальное число итераций MaxIterIRLS не достигнуто, то повторяем шаги 2–8 (iteratively reweighed least-squares),

10. End.

Задачи 2.3 и 2.4 решаются с помощью QR-разложения матриц $\mathbf{V}^{\mathrm{T}}\mathbf{W}_i\mathbf{V}$ и $\mathbf{U}^{\mathrm{T}}\mathbf{W}^j\mathbf{U}$ соответственно.

Авторы статьи [7], ссылаясь на проведенные численные эксперименты, предлагают выбрать $\sigma = 1.4826\,\mathrm{med}\,|R - \mathrm{med}\,|R||$, где R — это вектор, составленный из всех элементов матрицы остатков $\mathbf{R} = \{r_{ij}\}_{i,j=1}^{n,p}$, то есть

$$R = (r_{11}, \ldots, r_{1p}; r_{21}, \ldots r_{2p}; \ldots; r_{n1}, \ldots, r_{np}).$$

Параметр α предлагается взять равным 4.685. Также говорится, что максимальное количество итераций N_α и N_{IRLS}, необходимых для сходимости, достаточно взять 5 и 10 для достижения приемлемой точности.

Критерий сходимости:

$$\left\|\mathbf{W}^{1/2} \odot (\mathbf{Y} - \mathbf{U}\mathbf{V}^{\mathrm{T})}\right\|_F^2 \leq \varepsilon.$$

У данного метода присутствуют существенные недостатки. Описанный алгоритм из статьи не подходит, к примеру, для рядов с растущей амплитудой. По умолчанию остатки нормировались на σ, которая задавалась константой. Но в случае растущей амплитуды данная нормировка приводит к неправильной идентификации точек с выбросами. В таком случае в точках с большой амплитудой веса некорректно занижаются, и точки, не содержащие выбросов, могут получить вес, меньший чем у выбросов в начале ряда. Поэтому приходим к выводу, что нормирующий параметр необходимо задавать динамически. Будем рассматривать вариант с заменой числа σ на ряд, равный тренду ряда из модуля остатков.

Изменения σ оказывается недостаточным для того, чтобы сделать метод подходящим для рядов с растущей амплитудой. В методе имеется параметр α, который влияет на то, какие точки воспринимать как выбросы, а какие — нет. В классическом методе из статьи этот параметр также задается константой. Но из-за роста разброса остатков в точках без выбросов к концу ряда, многие из этих точек также получают искусственно заниженные веса. Попробуем исправить этот недостаток.

Пусть $R = (r_1, \ldots, r_q)^{\mathrm{T}}$ — вектор остатков, где $q = mn$. Если остатки $r_i \sim N(0, \sigma^2)$, то $|r_i| \sim N_H(\sigma^2)$, где $N_H(\sigma^2)$ — полунормальное распределение с параметром σ^2. Пусть $\mathbb{E}|r_i| = \mu$. Для полунормального распределения известны следующие формулы для среднего и дисперсии:

$$\mathbb{E}|r_i| = \sigma\sqrt{\frac{2}{\pi}}, \quad \mathbb{D}|r_i| = \sigma^2(1 - \frac{2}{\pi}) = C\mu^2, \quad \text{где } C = \text{const}.$$

В алгоритме при вычислении весов проводится сравнение $|\frac{R}{\sigma}|$ с константой α. Если модуль превосходит α, то веса обнуляются. Чем ближе значение модуля к α, тем ниже вес в данной точке. Но так как в конце ряда разброс остатков больше, чем в начале, то при задании α константой, многие точки получают маленькие веса. Поэтому необходимо либо задавать пороговое значение динамически, либо ввести дополнительное преобразование, чтобы отрегулировать разброс остатков в начале и конце ряда. Квадратичная связь дисперсии и среднего модуля остатков наталкивает на мысль брать логарифм модулей остатков перед тем, как проводить сравнение с α. Однако очень маленькие значения остатков могут привести к большим отрицательным значениям логарифма, поэтому попробуем для начала извлекать корень из ряда модулей остатков.

Таким образом, получаем модификацию алгоритма, подходящую для рядов с быстрорастущей (или убывающей) амплитудой. Данная модификация отличается от исходного алгоритма шагами 4 и 5.

4.a Ганкелизация матрицы $\mathbf{R}$ и получение ряда длины N из остатков: $\mathsf{R} = \mathcal{T}^{-1}\Pi_{\mathcal{H}}(\mathbf{R})$,

4.b Вычисление $\sigma = (\sigma_1, \ldots, \sigma_N)$ как тренд из ряда $|\mathsf{R}|$,

4.c Вычисление ряда $|\sigma^{-1}\mathsf{R}|$ и получение матрицы $\mathbf{R}^* = \{r^*_{ij}\}_{i,j=1}^{n,p} = \mathcal{T}(|\sigma^{-1}\mathsf{R}|)$,

5. Вычисление матрицы весов $\mathbf{W}$, используя функцию $w(x) = \frac{\partial\rho(x)}{\partial|x|}\frac{1}{|x|}$:

$$w(x) = \begin{cases} (1 - (\frac{|x|}{\alpha})^2)^2, & |x| \le \alpha \\ 0, & |x| > \alpha \end{cases},$$

где $x = \sqrt{r^*_{ij}}$.

Посмотрим на график весов, чтобы убедиться, что только точки, содержащие выбросы, получили маленькие веса. На рисунках 2.1 и 2.2 изображен график ряда с 5%

выбросов и веса, получившиеся в результате применения модификации метода. На графике весов видно, что точки, в которых содержались выделяющиеся наблюдения, получили нулевые веса. В остальных точках веса колеблются от 0.8 до 1.

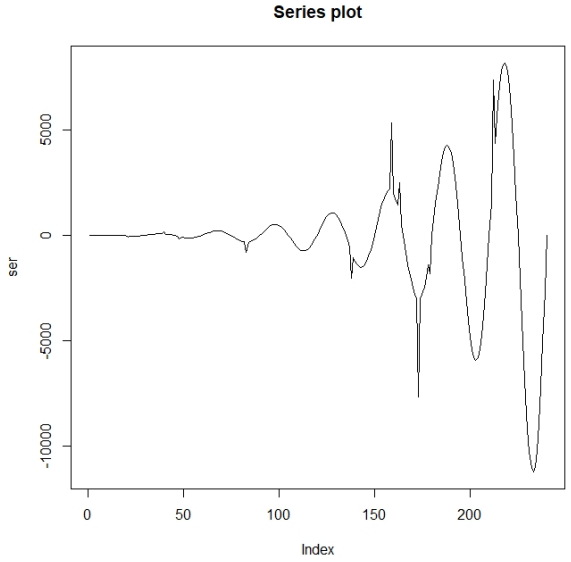

Рис. 2.1. График ряда с 5% выбросов.

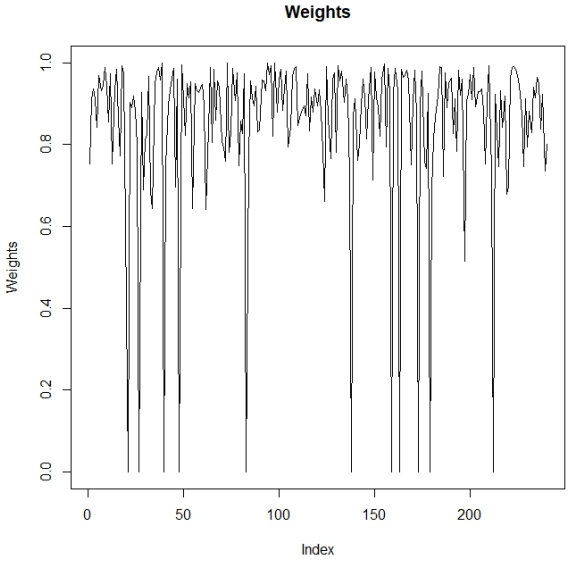

Рис. 2.2. Веса.

2.5. Оценка трудоемкости методов

Вопросу трудоемкости L1-SSA уделялось большое внимание во многих работах, посвященных построению $\mathbb{L}_1$-проекции на множество матриц ранга, не превосходящего r. В статье [13] приведен алгоритм, решающий точно эту задачу. Пусть имеется вещественная траекторная матрица размерности $n \times p$ ранга r. В случае $n > p$ метод имеет трудоемкость $O(2^p)$. Трудоемкость в случае $n < p$, представляющем больший интерес в случае временных рядов, составляет $O(p^r)$. В данной работе в разделе 2.2 мы рассматривали методы, решающие эту задачу с меньшей точностью, но более эффективно.

Сравним теоретические трудоемкости описанных алгоритмов.

Метод, использующий взвешенную медиану

Вычислим порядок операций, требующихся для построения проекции матрицы размерности $n \times p$ на множество матриц ранга, не превосходящего r, методом из раздела 2.2.1. Трудоемкость нахождения медианы составляет $O(N)$, где N — объем выборки.

То есть на внутренний цикл необходимо np операций. Алгоритм находит все p собственных троек. Таким образом, порядок операций составляет

$$\mathrm{T}_{\mathrm{robustSvd}} = O(p^2 n N_{iter}), \tag{2.5}$$

где N_{iter} — число итераций, необходимых для сходимости. Число итераций не превышает максимального числа итераций, которое можно задать константным, не зависящим от n и p.

Последовательный метод

Вычислим трудоемкость последовательного алгоритма из раздела 2.2.2. Трудоемкость составляет $O((pP_1 + P_2)N_{iter})$, где P_1 и P_2 — трудоемкость решения задач линейного программирования, а N_{iter} — общее количество итераций для сходимости метода, которое также считаем не зависящим от n, p, r. Согласно [14], сложность вычисления задачи линейного программирования с v переменными и c ограничениями составляет $O(c \log v)$. В статье [6] вычислено количество переменных и ограничений в этих задачах, и получено, что трудоемкость может быть оценена как

$$\mathrm{T}_{\mathrm{l1pca}} = O(np \log(2pn + nr) N_{iter}), \tag{2.6}$$

Взвешенный метод наименьших квадратов

Теоретическая трудоемкость метода из раздела 2.4 составляет, согласно статье [7],

$$\mathrm{T}_{\mathrm{IRLS}} = O(npr^2 N_\alpha N_{IRLS}), \tag{2.7}$$

где N_α и N_{IRLS} — общее количество итераций для решения задач (2.3), (2.4) и сходимости взвешенного метода наименьших квадратов с обновлением весов. Количество итераций мы брали постоянными и не зависящими от n, p и r. При подсчете трудоемкости авторы статьи [7] используют книгу [15], в которой приводятся эффективные алгоритмы QR-разложения матрицы.

Сравнение трудоемкостей

Сравним теоретические трудоемкости последовательного метода и взвешенного метода наименьших квадратов. Необходимо сравнить (2.6) и (2.7). Рассмотрим 2 случая. Задача сводится к сравнению $\log(n(2p + r))$ и r^2.

Таблица 2.1. Время работы программы и число итераций для различных методов для $M = 10$ реализаций ряда.

	l1pca	robustSvd	IRLS
time	54 sec	237 sec	39 sec
N_{iter}	10	10	5*10

Пусть n фиксировано, маленькое, а $p \sim N$. Это соответствует случаю, когда длина окна маленькая, и траекторная матрица вытянута. В таком случае только при N порядка 10^{r^2} и больше трудоемкость последовательного метода оказывается хуже. Но такая большая длина ряда маловероятна, поэтому можно сделать вывод, что в таком случае последовательный метод должен работать быстрее.

Пусть траекторная матрица ряда близка к квадратной, то есть $n \sim N/2,\ p \sim N/2$. Тогда, если поводить сравнение, то получаем, что надо сравнить $\frac{N}{2}(N + r)$ и 10^{r^2}. Вычислив корни квадратного уравнения, получаем, что при N порядка, меньшего $\sqrt{10^{r^2}}$, трудоемкость последовательного метода оказывается меньше.

Можно сделать вывод, что теоретическая трудоемкость последовательного метода оказывается меньше теоретической трудоемкости взвешенного метода наименьших квадратов.

Сравним теоретические результаты со временем работы методов на конкретном примере. Рассмотрим ряд, являющийся суммой экспоненты и синуса с гауссовским шумом. Пусть длина ряда $N = 240$.

Для того, чтобы иметь возможность сравнить время работы методов, необходимо критерии остановки сделать такими, чтобы методы выдавали примерно одинаковые по точности результаты. Однако это оказывается нетривиальной задачей, поэтому поставим максимальное количество итераций для каждого метода такие, чтобы точность оказывалась приемлемой для каждого из методов. Сравним время работы, учитывая количество итераций.

В таблице 2.1 приведено время работы четырех методов с 1% выбросов для 10 реализаций ряда. Время работы метода, использующего взвешенную медиану, намного больше времени работы остальных двух методов. Методы l1pca и IRLS для данного ряда с маленьким рангом работают примерно одинаково по времени.

Глава 3

Вычислительные эксперименты

Сравним результаты работы описанных методов на двух примерах.

3.1. Пример 1

Для начала рассмотрим пример из работы [4], но добавим большее количество выбросов (1% и 5%) в случайных точках ряда. Проверим, какой из приведенных алгоритмов окажется наиболее устойчивым.

Длина ряда $N = 240$. Рассмотрим временной ряд

$$f_n = e^{n/N} + \sin(2\pi n/120 + \pi/6) + \varepsilon_n,\ \varepsilon_n \sim N(0,1).$$

На рис. 3.1 изображен график ряда при 1% выбросов с величиной выброса $5y_i$.

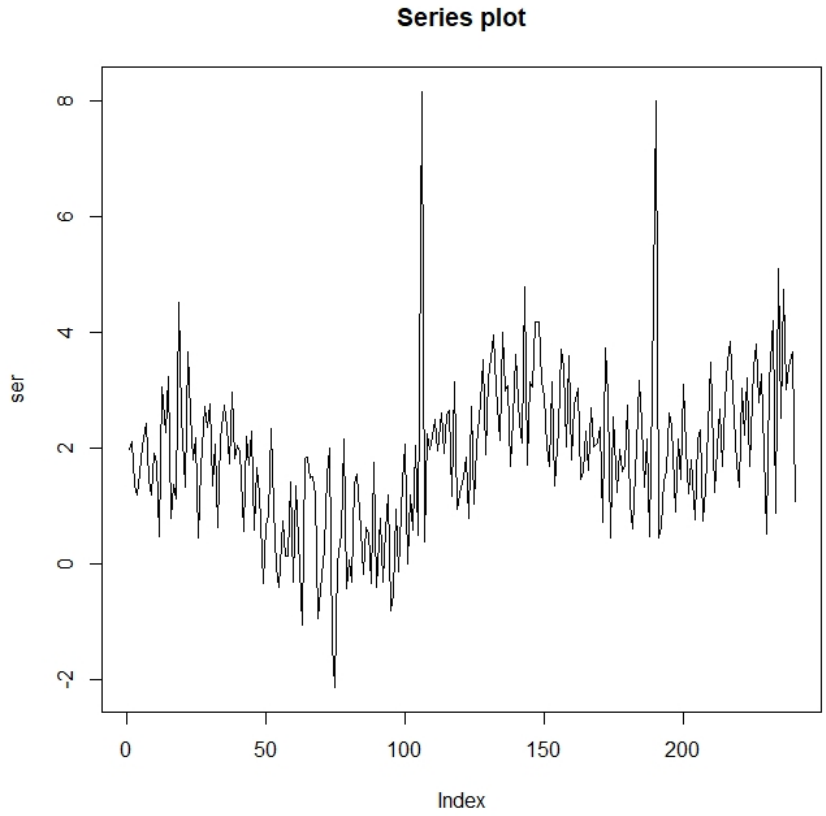

Рис. 3.1. График ряда при 1% выбросов с величиной выброса $5y_i$.

Сравнение будет проводиться по величине среднеквадратичной ошибки, согласованной с $\mathbb{L}_2$, которая вычисляется по формуле

$$\mathrm{MSE} = \mathbb{E}\left(\frac{1}{N}\sum_{i=1}^{N}(s_i - \hat{s}_i)^2\right), \tag{3.1}$$

где S — сигнал, $\hat{S}$ — его оценка. Будем вычислять

$$\mathrm{RMSE} = \sqrt{\mathrm{MSE}},$$

а также будем сравнивать методы по величине ошибки, согласованной с $\mathbb{L}_1$, которая имеет вид

$$\mathrm{MAD} = \mathbb{E}\left(\frac{1}{N}\sum_{i=1}^{N}|s_i - \hat{s}_i|\right). \tag{3.2}$$

Возьмем количество реализаций ряда $M = 10$. Будем находить оценки математических ожиданий (3.1) и (3.2), а далее из оценки MSE будем извлекать корень, получая RMSE.

Для начала попробуем сравнить работу методов без шума и без выделяющихся наблюдений. В таком случае стандартный метод L2-SSA, метод с весами IRLS и последовательный метод l1pca выдают нулевую ошибку с точностью до погрешности вычислений. Оценка ошибки восстановления сигнала без шума и выбросов методом, использующим взвешенную медиану, не равна нулю (около 0.043). Это доказывает, что метод решает не ту задачу, которую нам необходимо решить. Поэтому этот метод рассматривать не будем.

В таблицах 3.1 и 3.2 представлены результаты сравнения для четырех методов. Выброс добавлялся заменой значения y_i на $y_i + 5y_i$.

Ранг ряда равен 3. Во всех методах берется длина окна $L = 120$, равная половине длины ряда, и восстановление сигнала ведется по 3 компонентам.

Первая строка таблиц соответствует стандартному методу SSA с большой длиной окна ($L = 120$). Вторая строка — метод l1pca из пакета pcaL1 [11] (соответствует последовательному методу из раздела 2.2.2). Третья строка соответствует стандартному методу из раздела 2.4 с использованием взвешенного метода наименьших квадратов. Четвертая строка соответствует модификации взвешенного метода наименьших квадратов.

При сравнении методов со стандартным использовался пакет Rssa [16].

Таблица 3.1. Оценки RMSE для различных методов для $M = 10$ реализаций ряда.

Method	0%	1%	5%
Basic SSA	0.181	0.238	0.607
l1pca	0.204	0.249	0.242
IRLS	0.182	0.193	0.194
IRLS (modif.)	0.182	0.191	0.230

Таблица 3.2. Оценки MAD для различных методов для $M = 10$ реализаций ряда.

Method	0%	1%	5%
Basic SSA	0.149	0.191	0.484
l1pca	0.166	0.188	0.183
IRLS	0.146	0.151	0.160
IRLS (modif.)	0.145	0.151	0.178

При отсутствии выбросов наиболее точным все так же остается классический метод SSA. Но заметим, что метод IRLS при отсутствии выделяющихся наблюдений работает не сильно хуже. Он же остается наиболее устойчивым в случае появления выбросов. Можно заметить, что модификация метода IRLS при 5% выбросов для данного ряда без растущей амплитуды дает результат немного хуже, чем стандартный IRLS.

Итак, можно сделать вывод, что метод IRLS для рассмотренного ряда работает достаточно точно при отсутствии выбросов, а также является наиболее устойчивым к выделяющимся наблюдениям среди остальных рассмотренных методов.

Проверка значимости сравнения

Проверим значимость сравнения по критерию для зависимых выборок. Проверим гипотезу, что MSE для некоторых методов равны между собой.

$\mathrm{H}_0 : \mathbb{E}(\xi_1 - \xi_2) = 0$. Имеем две выборки $X = (x_1, \ldots, x_M)$ и $Y = (y_1, \ldots, y_M)$ объема M. Обозначим $\bar{X}$ и $\bar{Y}$ — их выборочные средние, s_x^2 и s_y^2 — выборочные дисперсии, $\hat{\rho}$ — коэффициент корреляции. Статистика критерия

$$t = \frac{\sqrt{M}(\bar{X} - \bar{Y})}{\sqrt{s_x^2 + s_y^2 - 2 s_x s_y \hat{\rho}}}.$$

имеет асимптотически нормальное распределение. Критерий является двухсторонним.

Таблица 3.3. P-value для сравнения MSE методов l1pca и IRLS в зависимости от количества выбросов

	0%	1%	5%
l1pca	0.204	0.249	0.242
IRLS	0.182	0.193	0.194
p-value	0.115	0.0079	0.018

Так как наилучшие результаты получились при использовании методов l1pca и IRLS, то проверим, является ли отличие между этими методами значимым. В таблице 3.3 приведены p-value для сравнения среднеквадратичных ошибок для последовательного метода l1pca и метода с весами IRLS. При уровне значимости 0.05 все сравнения в присутствии выбросов оказываются значимыми. Без выделяющихся наблюдений — не значимы.

3.2. Пример 2

Попробуем рассмотреть непохожий на предыдущий пример ряд, у которого будет достаточно большой разброс значений, и исследуем устойчивость методов.

Рассмотрим пример, предложенный в статье [5], и проведем для этого примера вычислительный эксперимент.

Пусть длина ряда $N = 240$. Рассмотрим временной ряд

$$f_n = ne^{4n/N} \sin(2\pi n/120) + \varepsilon_n,\ \varepsilon_n \sim N(0,1).$$

Ранг ряда равен 4. У такого ряда разброс собственных значений очень велик. Это может приводить к тому, что некоторые компоненты сигнала могут смешиваться с шумом. Однако шум рассматриваемого размера не портит отделимость сигнала от шума. Выбросы будут находиться в случайно выбранных точках ряда. Сравнение будем проводить при 1% и 5% выбросов, а также без выделяющихся наблюдений. В случайно выбранных точках y_i значение будет заменяться на $y_i + 1.5y_i$.

На рис. 3.2 изображен график ряда при 1% выбросов с величиной выброса $1.5y_i$.

Результаты сравнения методов при различном проценте выделяющихся наблюдений представлены в таблицах 3.4 и 3.5.

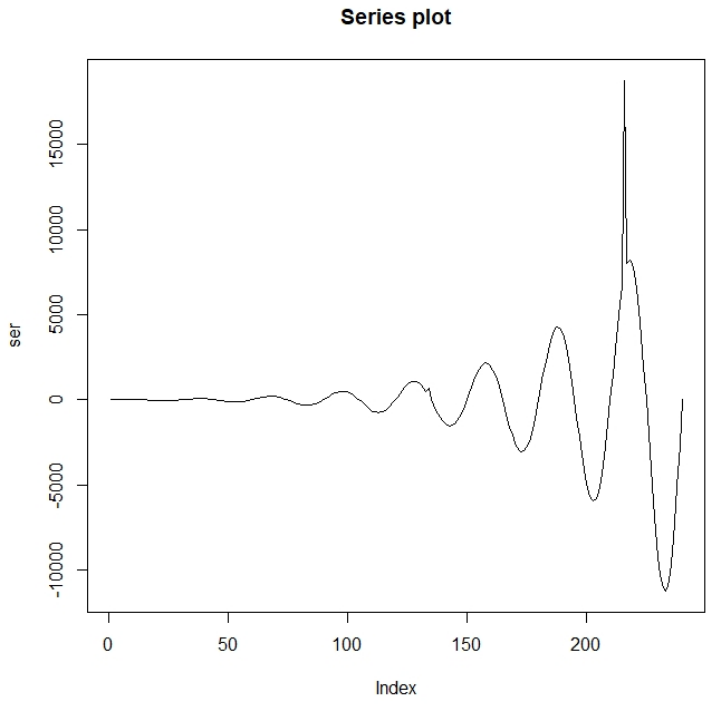

Рис. 3.2. График ряда при 1% выбросов с величиной выброса $1.5y_i$.

Таблица 3.4. Оценки RMSE для различных методов для $M = 10$ реализаций ряда.

Method	0%	1%	5%
Basic SSA	0.21	213.9	537.1
l1pca	0.26	7.3	16.5
IRLS	0.23	485.1	813.9
IRLS (modif.)	0.22	9.2	11.1

Метод IRLS при такой быстрорастущей амплитуде ряда работает плохо, однако его модификация достаточно хорошо справляется с восстановлением сигнала как при отсутствии выбросов, так и при большом их количестве. Можно сделать вывод, что как минимум для рассмотренного нами примера, модификация метода IRLS работает хорошо по сравнению с остальными методами.

Посмотрим на осмысленность восстановления последовательным методом и методом взвешенных наименьших квадратов. На рисунке 3.3 изображены исходный сигнал и восстановление методом l1pca и IRLS. Видно, что восстановление и тем, и другим методами довольно осмысленно и практически не отличается друг от друга (два графика практически полностью наложились друг на друга). Выбросы обрезались хорошо и

Таблица 3.5. Оценки MAD для различных методов для $M = 10$ реализаций ряда.

Method	0%	1%	5%
Basic SSA	0.15	34.4	192.5
l1pca	0.19	2.1	4.39
IRLS	0.16	18.0	56.1
IRLS (modif.)	0.15	2.6	3.29

восстановленный сигнал совпал с исходным.

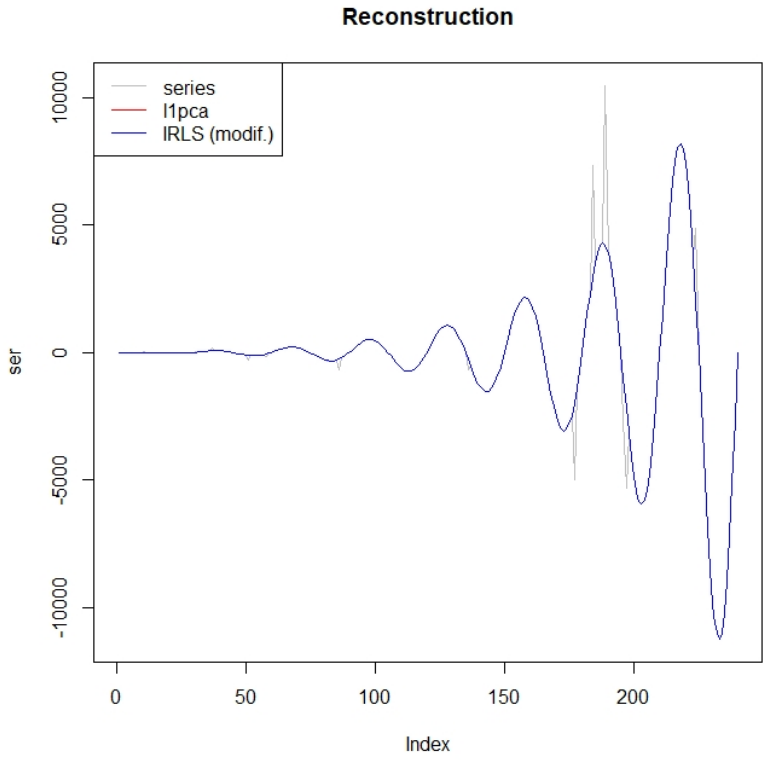

Рис. 3.3. График восстановленного ряда двумя методами при 5% выбросов с величиной выброса $1.5y_i$.

Для того, чтобы проверить, что восстановление сигнала описанными методами осмысленно, попробуем рассмотреть сам ряд, включая выбросы, как оценку сигнала и посчитать среднее отклонение от истинного сигнала. Ошибка в таком случае должна получиться больше, чем оценки ошибок методов. Результаты представлены в таблице 3.6. Действительно, оценки ошибок восстановления сигнала рассматриваемыми методами оказались меньше, чем ошибка, если в качестве оценки сигнала рассматривать исходный ряд.

Таблица 3.6. RMSE и MAD для исходного ряда в качестве оценки сигнала.

	0%	1%	5%
RMSE	0.97	542.9	953.8
MAD	0.77	39.1	114.7

Таблица 3.7. P-value для сравнения MSE методов l1pca и IRLS (modif.) в зависимости от количества выбросов

	0%	1%	5%
l1pca	0.26	7.3	16.5
IRLS (modif.)	0.22	9.2	11.1
p-value	0.0018	0.17	0.35

Проверка значимости сравнения

В таблице 3.7 приведены p-value для сравнения среднеквадратичных ошибок для последовательного метода l1pca и модифицированного метода с весами IRLS (modif.). При уровне значимости 0.05 и отсутствии выбросов сравнения оказываются значимыми. В присутствии выделяющихся наблюдений значимых отличий между этими двумя методами нет.

3.3. Пример 3

Основное отличие первого и второго примера состояло в том, что у второго ряда был сильный рост амплитуды. Однако у первого ряда присутствовал растущий экспоненциальный тренд, что могло также повлиять на работу методов. Попробуем рассмотреть простой стационарный ряд, у которого не растет амплитуда, и сравним методы на таком примере. Длина ряда по-прежнему $N = 240$. Рассмотрим ряд

$$f_n = \sin(2\pi n/120 + \pi/6) + \varepsilon_n,\ \varepsilon_n \sim N(0,1).$$

График ряда при 1% выбросов с величиной выброса $5y_i$ изображен на рис. 3.4.

Результаты сравнения представлены в таблицах 3.8 и 3.9. Метод IRLS оказывается для этого ряда устойчивее, чем последовательный метод.

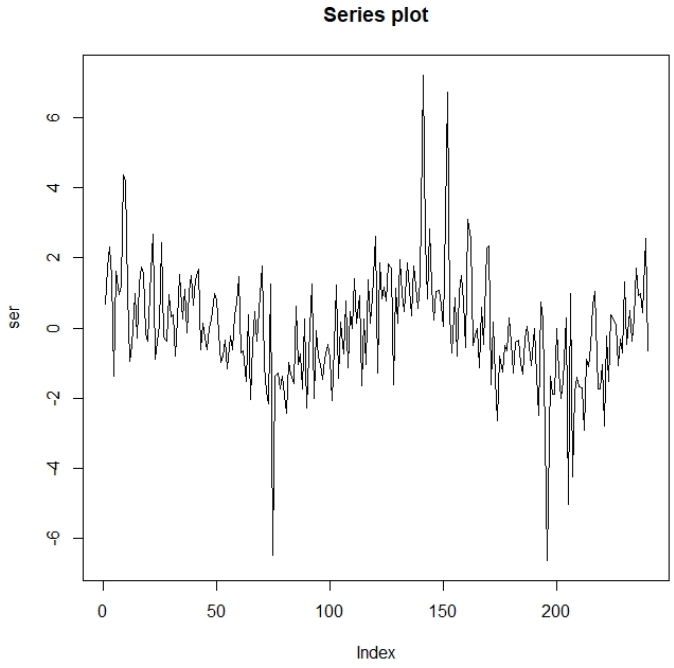

Рис. 3.4. График ряда при 1% выбросов с величиной выброса $5y_i$.

Таблица 3.8. Оценки RMSE для различных методов для $M = 10$ реализаций ряда.

Method	0%	1%	5%
Basic SSA	0.162	0.176	0.327
l1pca	0.194	0.231	0.230
IRLS	0.164	0.165	0.195
IRLS (modif.)	0.164	0.167	0.240

Проверка значимости сравнения

В таблице 3.10 приведены p-value для проверки значимости отличия последовательного метода и метода с весами. Как и в примере из пункта 3.1, при отсутствии выбросов при уровне значимости 0.05 отличие не значимо, а с выбросами — значимо.

3.4. Выводы

Исходя из полученных результатов сравнений методов, можем сделать следующие выводы.

При отсутствии выбросов и шума нулевую ошибку с точностью до погрешности вычислений дают только стандартный метод L2-SSA, последовательный метод l1pca и

Таблица 3.9. Оценки MAD для различных методов для $M = 10$ реализаций ряда.

Method	0%	1%	5%
Basic SSA	0.132	0.147	0.250
l1pca	0.150	0.185	0.184
IRLS	0.133	0.134	0.159
IRLS (modif.)	0.134	0.135	0.185

Таблица 3.10. P-value для сравнения MSE методов l1pca и IRLS в зависимости от количества выбросов

	0%	1%	5%
l1pca	0.194	0.231	0.230
IRLS	0.164	0.165	0.195
p-value	0.105	0.003	0.002

IRLS. Ошибка метода robustSvd достаточно большая, то есть метод не решает необходимую нам задачу.

Хорошими оказались два метода: последовательный метод и взвешенный метод наименьших квадратов, включая его модификацию.

Для ряда, амплитуда которого остается постоянной, наилучшим методом при отсутствии выбросов остается стандартный L2-SSA, однако значение ошибки метода IRLS при отсутствии выбросов оказывается ненамного больше. Метод IRLS является для такого примера наиболее устойчивым.

Для ряда с растущей амплитудой наиболее устойчивыми методами оказываются l1pca и IRLS.

Теоретическая трудоемкость последовательного метода $O(np\log(2pn + nr)N_{iter})$ оказывается меньше теоретической трудоемкости метода с весами, которая составляет $O(npr^2N_\alpha N_{iter})$. Однако на практике время работы этих двух методов небольшое и не сильно отличающееся друг от друга: 54 и 39 секунд.

Заключение

В работе были приведены и исследованы некоторые варианты модификации метода SSA с целью повышения устойчивости к выбросам.

В главе 1 были введены основные понятия и обозначения, описан алгоритм базового метода.

В главе 2 была введена общая схема метода с проекторами на пространство ганкелевых матриц и множество матриц ранга, не превосходящего r, без указания конкретной нормы. Для построения $\mathbb{L}_1$-проектора на множество матриц ранга, не превосходящего r, было предложено два способа. Первый из них был взят из статьи [5] и использует взвешенную медиану. Второй метод последовательный, который уже реализован в R-пакете [11]. Главное отличие этих методов состоит в том, что первый метод ищет собственные тройки по очереди, но не в порядке убывания собственных чисел. Второй метод ищет все компоненты одновременно в виде матрицы.

Также была рассмотрена другая идея построения устойчивого метода — присвоение точкам, содержащим выбросы, меньший вес. Этой идее соответствует метод из статьи [7]. Но так как было обнаружено, что этот метод не подходит для рядов с растущей или убывающей амплитудой, была разработана модификация данного метода.

Бало проведено сравнение теоретических трудоемкостей методов. Рассмотрено два случая: случай с квадратной и вытянутой траекторной матрицей. В обоих случаях теоретическая трудоемкость последовательного метода оказалась наименьшей.

Глава 3 была посвящена численным экспериментам. Классический метод L2-SSA сравнивался с тремя вариантами L1-SSA (l1pca, robustSvd и IRLS). В случайных точках ряда y_i добавлялись выделяющиеся наблюдения, равные $5y_i$ для первого примера и $1.5y_i$ для второго. Для начала проводилось сравнение методов без выделяющихся наблюдений, а затем при 1% и 5% выбросов. Сравнение проводилось по величине оценок ошибок RMSE и MAD.

Оказалось, что два метода работают достаточно хорошо — это последовательный метод l1pca и метод с весами IRLS, а также его модификация.

Сначала рассматривался пример из работы [4], но с добавлением большего количества выбросов. Было показано, что в таком случае метод IRLS оказался довольно точным и устойчивым.

Далее ряд был заменен на ряд с большим разбросом значений. На таком примере преимущество метода IRLS пропадает. Однако его модификация оказывается устойчивой и точной при отсутствии выбросов. Последовательный метод также показал хорошие результаты.

Таким образом, если ряд стационарный, то наилучшим методом оказывается метод, сопоставляющий точкам, содержащие выбросы, меньший вес. В случае нестационарных рядов получилось построить модификацию, которая, как минимум для рассмотренного примера, работает неплохо.

Список литературы

1. Advanced spectral methods for climatic time series / M. Ghil, R. M. Allen, M. D. Dettinger et al. // Reviews of Geophysics. — 2002. — Vol. 40, no. 1. — P. 1–41.
2. Broomhead D., King G. Extracting qualitative dynamics from experimental data // Physica D. — 1986. — Vol. 20. — P. 217–236.
3. Vautard M., Ghil M. Singular spectrum analysis in nonlinear dynamics, with applications to paleoclimatic time series // Physica D. — 1989. — Vol. 35. — P. 395–424.
4. Третьякова А. Л. Устойчивые варианты метода SSA для анализа временных рядов: выпускная квалификационная работа, науч.рук. к.ф.-м.н., доцент Голяндина Н.Э. — 2018.
5. Rodrigues P., Lourenco V., Mahmoudvand R. A robust approach to singular spectrum analysis // Quality and Reliability Engineering International. — 2018. — 06. — Vol. 34.
6. Brooks J. P., Jot S. pcaL1: An implementation in R of three methods for L1-norm principal component analysis. — 2013.
7. Chen K., Sacchi M. Robust reduced-rank filtering for erratic seismic noise attenuation // GEOPHYSICS. — 2015. — 01. — Vol. 80. — P. V1–V11.
8. Голяндина Н. Э. Метод «Гусеница»-SSA: анализ временных рядов: Учеб. пособие. — Санкт-Петербург : ВВМ, 2004.
9. Stacklies W., Redestig H., Scholz M. et al. — pcaMethods – a Bioconductor package providing PCA methods for incomplete data, 2007. — R package version 1.72.0. Access mode: `https://bioconductor.org/biocLite.R`.
10. pcaMethods – a Bioconductor package providing PCA methods for incomplete data / Wolfram Stacklies, Henning Redestig, Matthias Scholz et al. // Bioinformatics. — 2007. — Vol. 23. — P. 1164–1167.
11. Jot S., Brooks J. P., Visentin A., Park Y. W. — pcaL1: L1-Norm PCA Methods, 2017. — R package version 1.5.2.
12. Hawkins D. M., Liu L., Young S. S. Robust singular value decomposition technical report number 122, National Institute of Statistical Sciences 19. — 2002.
13. Markopoulos P., Karystinos G., Pados D. Optimal algorithms for L1-subspace signal processing // IEEE Transactions on Signal Processing. — 2014. — 10. — Vol. 62. — P. 5046–5058.

14. Chvátal V. Linear Programming. Series of books in the mathematical sciences. — New York : W.H. Freeman Company, 1983. — ISBN: 9780716711957.
15. Golub G. H., Van Loan C. F. Matrix Computations. — Third edition. — The Johns Hopkins University Press, 1996.
16. Korobeynikov A., Shlemov A., Usevich K., Golyandina N. — Rssa: A collection of methods for singular spectrum analysis, 2016. — R package version 1.0. Access mode: http://CRAN.R-project.org/package=Rssa.