

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

На правах рукописи

Нина Эдуардовна Голяндина

**ОБЩИЙ ПОДХОД К ТЕОРИИ И МЕТОДОЛОГИИ МЕТОДА
АНАЛИЗА СИНГУЛЯРНОГО СПЕКТРА**

Научная специальность

1.2.2. Математическое моделирование, численные методы и комплексы программ

Диссертация представлена на соискание ученой степени доктора
физико-математических наук

Перевод с английского языка

Санкт-Петербург — 2023

Оглавление

Введение	7
Используемые обозначения	16
Глава 1. Общая структура методов SSA-семейства	17
1.1. Временные ряды и цифровые изображения: общие проблемы	18
1.2. Общая схема семейства SSA и основные понятия	20
1.2.1. Общий метод SSA	20
1.2.2. Основные понятия	22
1.3. Различные версии SSA	25
1.3.1. Разложение \mathbf{X} в сумму матриц ранга один	26
1.3.2. Версии SSA, работающие с различными формами объекта	29
1.4. Разделимость в SSA	30
1.5. Прогнозирование, интерполяция, аппроксимация малого ранга и оценка параметров в SSA	31
1.6. Сравнение SSA с другими методами	32
1.6.1. Преобразование Фурье, фильтрация, шумоподавление	33
1.6.2. Параметрическая регрессия	34
1.6.3. ARIMA и ETS	35
1.7. Библиографические заметки	36
1.7.1. Некоторые недавние применения SSA	37
1.7.2. SSA для предварительной обработки / комбинация методов	38
1.8. Basic SSA для одномерных временных рядов	39
1.8.1. Метод	39
1.8.2. Модель временных рядов	41
1.8.3. Разделимость и выбор параметров	43
1.8.4. Разложение экспоненциально-модулированных гармоник	45
1.8.5. Алгоритм	46
1.8.6. Модификация Toeplitz SSA для стационарных временных рядов	47
1.8.7. Пример идентификации и разложения	49
1.8.8. Пример проблем с разделимостью	51
1.9. Прогнозирование и оценка параметров для одномерных временных рядов	54
1.9.1. Выделение сигнала через проекции	54
1.9.2. Оценка параметров	55
1.9.3. Прогнозирование	58

Глава 2. Шаг разложения для одномерных временных рядов	64
2.1. SSA с проекцией	64
2.1.1. SSA с центрированием	65
2.1.2. SSA с проекцией	65
2.1.3. Примеры	72
2.1.4. Обзор результатов	78
2.2. Iterative Oblique SSA	79
2.2.1. Oblique SVD и SSA	79
2.2.2. Метод	87
2.2.3. Алгоритмы	90
2.2.4. Пример. Разделимость синусоид с близкими частотами	92
2.3. Filter-adjusted O-SSA и SSA с производными	94
2.3.1. SSA с производными. Вариация для сильной разделимости	95
2.3.2. Filter-adjusted O-SSA	98
2.3.3. Примеры	100
2.4. SSA-ICA	107
2.4.1. Максимизация энтропии	107
2.4.2. SOBI-AMUSE	110
2.5. Автоматическая идентификация	118
2.5.1. Метод низких частот для идентификации тренда	119
2.5.2. Частотный метод идентификации колебательной компоненты	120
2.5.3. Метод идентификации колебательной компоненты по регулярности углов	123
2.5.4. Сравнение методов для идентификации гармоники	127
Глава 3. Модельные задачи: заполнение пропусков, оценка сигнала, обнаружение сигнала	130
3.1. Метод SSA заполнения пропусков на основе подпространств	130
3.1.1. Обзор методов SSA для заполнения пропусков	130
3.1.2. Вспомогательные утверждения	136
3.1.3. Запаздывающие векторы и траекторные пространства временных рядов конечного ранга с отсутствующими данными	139
3.1.4. Поиск траекторных пространств исходного временного ряда и его аддитивных компонентов	140
3.1.5. Комментарии к реализации подпространственного метода заполнения пропусков	142
3.2. Возможность построения весов в задаче HSLRA	145
3.2.1. Свертка матриц и векторов	147
3.2.2. Связь между векторной и матричной формами HSLRA	148

3.2.3.	Производящие функции и свертка ленточных матриц	149
3.2.4.	Автоковариационные матрицы и их инверсии для AR(1) и AR(2) моделей	151
3.2.5.	Изучение существования решений задачи слепой деконволюции для матриц, пропорциональных обратным ковариационным матрицам в авторегрессионных моделях	153
3.2.6.	Несуществование деконволюции $\mathbf{W} = \mathbf{A} * \mathbf{B}$ для стационарной AR(p) модели с общим p и диагональю \mathbf{B}	160
3.3.	Обнаружение сигналов с помощью сингулярного спектрального анализа Монте-Карло: Множественное тестирование	163
3.3.1.	Статистический подход к проверке гипотез	164
3.3.2.	Монте-Карло SSA	168
3.3.3.	Численное исследование	173
3.4.	Выбор параметров	181
3.4.1.	Введение	181
3.4.2.	Сигнальное подпространство	182
3.4.3.	Извлечение сигнала	184
3.4.4.	Рекуррентный прогноз SSA	189
3.4.5.	Подпространственные методы оценки параметров	191
3.4.6.	Скорость сходимости	192
3.4.7.	Выбор длины окна и разделимость	197
3.4.8.	SSA обработка стационарных временных рядов	201
3.4.9.	SVD-источники SSA и выбор параметров SSA	205
Глава 4. SSA для многомерных временных рядов	207
4.1.	MSSA анализ	208
4.1.1.	Метод	208
4.1.2.	Алгоритм	212
4.2.	Элементы теории MSSA	212
4.2.1.	Разделимость	212
4.2.2.	Ранги и подпространства	213
4.2.3.	Разложение э.-м. гармоник	216
4.2.4.	Комментарии по сравнению 1D-SSA и MSSA	218
4.3.	MSSA прогнозирование	219
4.3.1.	Метод	220
4.3.2.	Быстрый алгоритм векторного прогнозирования	224
4.3.3.	Пример моделирования: численное сравнение	226
4.4.	Автоматизация группировки в MSSA	229

Глава 5. Многомерный и Shaped SSA	235
5.1. Shaped 2D-SSA	236
5.1.1. Метод	236
5.1.2. Ранг фигурных массивов	238
5.1.3. Алгоритм	239
5.2. Частные случаи Shaped SSA	240
5.2.1. Shaped 1D-SSA	240
5.2.2. MSSA	241
5.2.3. 2D-SSA	244
5.2.4. M-2D-SSA	247
5.2.5. Комментарии к n D расширениям	248
5.3. Пример Shaped SSA	250
Глава 6. Пакет RSSA	253
6.1. Краткое введение в RSSA	254
6.2. Эффективность реализации	255
6.2.1. Эффективность R-пакета RSSA	255
6.2.2. Пример вычислений в RSSA	256
6.3. Унифицированный подход для реализации схемы SSA	256
6.3.1. Основной анализ	256
6.3.2. Различные модификации шага разложения	258
6.3.3. Методы, основанные подпространстве сигнала	258
Глава 7. Приложения к анализу реальных данных	259
7.1. Прогнозирование временных рядов ЕОР с помощью сингулярного спектрального анализа	259
7.1.1. Источники данных	261
7.1.2. Автоматический выбор параметров	263
7.1.3. Прогнозы на тестовом периоде	264
7.1.4. Выводы	265
7.2. Применение SSA к оценке плотности	267
7.2.1. Новый подход к оценке плотности	267
7.2.2. Качество оценок SSA	273
7.2.3. Приложение оценки функции распределения при исследовании рынка продаж	276
7.3. Двухэкспоненциальные модели паттернов экспрессии генов для зашумленных экспериментальных данных	278
7.3.1. Методы	279
7.3.2. Результаты и обсуждение	285

7.4. Shaped SSA для количественной оценки экспрессии генов, с применением к раннему эмбриону дрозофилы	288
7.4.1. Материалы	290
7.4.2. Выбор параметров, разделимость и идентификация компонентов	290
7.4.3. Остаточные периодические паттерны, создаваемые алгоритмами разделения сигналов	292
7.4.4. Выводы	299
7.5. Shaped 3D SSA для количественной оценки экспрессии генов, с применением к раннему эмбриону зебрафиш	302
7.5.1. Данные	305
7.5.2. Метод	305
7.5.3. Пример для модели расположении ядер на части сферы (“шапочке”) .	311
Заключение	317
Благодарности	321
Список литературы	322

Введение

Основным предметом исследования в данной работе являются временные ряды, а именно последовательность вещественных значений $\mathbf{X} = (x_1, \dots, x_N)$, где индекс обычно обозначает равноотстоящие моменты времени, однако может соответствовать и пространственным координатам. Анализ и прогноз временных рядов является очень распространённой задачей, так как огромное количество данных касается характеристик, меняющихся во времени. Также в работе изучаются системы временных рядов $\mathbf{X} = (\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(s)})$, где $\mathbf{X}^{(k)} = (x_1^{(k)}, \dots, x_N^{(k)})$, $k = 1, \dots, s$.

Кроме временных рядов в работе рассматриваются цифровые изображения в виде массива $\mathbf{X} = \{x_{ij}\}_{i,j=1}^{N_x, N_y}$. Самый общий вид исследуемых объектов — это $\mathbf{X} = \{x_{\mathbf{i}}\}_{\mathbf{i} \in \mathcal{N}}$, где \mathbf{i} — мультииндекс некоторой размерности, множество \mathcal{N} является линейно-упорядоченным и задает форму объекта.

Об исходном объекте предполагается, что он является суммой некоторых составляющих, например, это может быть сигнал и шум (случайный процесса с нулевым средним, не обязательно стационарный), зашумленная сумма синусоид, сумма тренда, периодичности и шума, сумма двух неслучайных компонент. Здесь под сигналом понимается неслучайная составляющая объекта, а под трендом/паттерном — неслучайная и медленно меняющаяся составляющая. Заметим, что часто для тренда рассматривается параметрическая модель, однако в данном случае, вообще говоря, таковая не предполагается. Задачами могут являться выделение, анализ и прогноз составляющих сигнала, заполнение пропусков, проверка гипотезы о (не)существовании сигнала во временном ряде.

Для решения описанных задач за основу был взят метод анализа сингулярного спектра (singular spectrum analysis, коротко SSA).

Алгоритм базовой модификации для анализа временного ряда выглядит так: на вход поступает вещественный временной ряд $\mathbf{X} = (x_1, \dots, x_N)$ длины N и длина окна L , $1 < L < N$. Метод SSA переводит временной ряд в траекторную матрицу размера $L \times K$, $K = N - L + 1$, оператором вложение \mathcal{T}_L :

$$\mathbf{X} = \mathcal{T}_L(\mathbf{X}) = \begin{pmatrix} x_1 & x_2 & \dots & x_K \\ x_2 & \dots & \dots & x_{K+1} \\ \dots & \dots & \dots & \dots \\ x_L & x_{L+1} & \dots & x_N \end{pmatrix}.$$

Затем рассматривается разложение матрицы в сумму элементарных матриц ранга 1, в базовом варианте это делается с помощью сингулярного разложения, после этого матрицы суммируются в m групп, после чего применением оператора \mathcal{T}_L^{-1} получаем результат: $\mathbf{X} = \tilde{\mathbf{X}}^{(1)} + \dots + \tilde{\mathbf{X}}^{(m)}$.

Таким образом, на вход поступает сумма объектов, а на выходе метода — оценки его слага-

гаемых. Кроме разложения исходного объекта результатом метода являются характеристики слагаемых, задающие их структуру, что позволяет прогнозировать неслучайные компоненты ряда, прогнозируя найденную структуру, заполнять в них пропуски, а также оценивать параметры временного ряда.

Пусть для простоты исходный объект состоит из двух интерпретируемых слагаемых $X = X^{(1)} + X^{(2)}$. Тогда результатом метода является разложение $X = \tilde{X}^{(1)} + \tilde{X}^{(2)}$. Возникают следующие вопросы: при каких условиях на слагаемые и параметры метода применяемый метод может выделить $X^{(1)}$ так, что $\tilde{X}^{(1)}$ равно $X^{(1)}$ точно или приближенно (условия *разделимости*), как при выполнении условий разделимости провести группировку элементарных компонент для выделения $X^{(1)}$, как проверить эти условия разделимости, как выглядит ошибка оценки сигнала в случае приближенного разделения и какие рекомендации можно дать по выбору параметров метода для уменьшения этой ошибки.

Понятие разделимости компонент временного ряда с помощью SSA было введено Некруткиным В.В. [1], что значительно продвинуло вперед теоретический подход к методу. Однако базовый вариант метода SSA имел недостатки с точки зрения разделимости компонент. Одной из целей данной работы является создание модификаций метода SSA, расширяющих класс разделимых компонент временного ряда. Хотя для теоретических результатов необходимо, чтобы компоненты временного ряда подчинялись некоторой модели, которую мы рассмотрим далее, применение этих результатов не требует того, чтобы компоненты ряда подчинялись этой модели. Поэтому модификации метода Basic SSA, который является непараметрическим, оставляют метод по-прежнему непараметрическим.

Хотя ряд других задач, например, прогноз и заполнение пропусков, требуют задания модели прогнозируемой составляющей временного ряда, ситуация примерно такая же, как при построении разложения временного ряда. А именно, теория строится для случая сигнала, удовлетворяющего модели, однако, например, метод прогнозирования применим и для случая, когда сигнал приближенно удовлетворяет модели.

Опишем модель на примере временного ряда. Пусть $S = (s_1, \dots, s_N)$ — сигнал (точнее, представляющая интерес компонента временного ряда). Устанавливаем длину окна L , $1 < L < N$; $K = N - L + 1$. Рассмотрим траекторную матрицу сигнала $S = \mathcal{T}_L(S)$. Пусть r обозначает ранг матрицы S .

Модель можно сформулировать в различных формах:

1. S — ганкелева матрица небольшого ранга $r < \min(L, K)$; в этом случае модель может быть параметризована на основе параметризации пространства $\text{colspace}(S)$ или ортогонального дополнения к нему. Такие временные ряды называются *временные ряды конечного ранга*.

2. Временной ряд управляемся некоторым линейным рекуррентным соотношением (ЛРС):

$$s_n = \sum_{k=1}^r a_k s_{n-k}, a_r \neq 0, n = r+1, \dots$$

Такие временные ряды называются *временными рядами, управляемые ЛРС*.

3. Временной ряд имеет явную параметрическую форму в виде конечной суммы:

$$s_n = \sum_k P_k(n) \exp(\alpha_k n) \sin(2\pi\omega_k n + \phi_k), \quad (1)$$

где $P_k(n)$ — многочлен от n , $\exp(\alpha_k n) = \rho_j^n$ за $\rho_j = e^{\alpha_j}$.

Первая модель носит более общий характер, однако при некоторых неограничительных условиях (например, для бесконечных временных рядов) эти три модели эквивалентны. Одним из недостатков третьего вида модели является то, что нужно задавать явный вид элемента сигнала, в то время как для первых двух вариантов достаточно задать только значение r .

В реальных задачах в рамках модели предполагается, что наблюдается зашумленный сигнал конечного ранга $X = S + R$, где S — сигнал конечного ранга, R — случайный шум с нулевым математическим ожиданием; шум не обязан быть стационарным. В этой модели методы типа SSA могут оценивать сигнал, оценивать параметры сигнала, заполнять пропуски, строить прогноз. Математические результаты относятся к построению методов прогноза и заполнения пропусков, к развитию алгебраического подхода к объектам конечного ранга, а также построению эффективных (быстрых и устойчивых) алгоритмов оценки сигнала конечного ранга.

Если сигнал имеет в точности конечный ранг (так называемый сигнал малого ранга), то решение задачи его оценивания строится через аппроксимацию ганкелевой матрицы и называется ганкелевой структурной аппроксимацией малого ранга (Hankel structured low-rank approximation, HSLRA). Для решения этой задачи можно использовать как запись модели в форме 1, так и запись модели в форме 2. В обоих случаях для рядов ранга r существует параметризация, отличающаяся от явной параметризации в форме 3. Эта задача решается как явно в виде взвешенного метода наименьших квадратов, так и в матричной форме 1. Большое количество результатов по HSLRA содержится в книге [2]. Метод Кэдзоу (Cadorz) [3] был предложен для решения задачи в матричной форме 1 независимо от развития SSA. Интересно, что базовый метод SSA как метод оценки сигнала совпадает с одной итерацией метода Кэдзоу. Это оказывается ключевым отличием, так как итерации Кэдзоу годятся только для случая сигналов конечного ранга, тогда как метод SSA гораздо более гибкий.

Наибольшую распространённость получил метод SSA для обработки одномерных временных рядов. Однако идеи метода оказались востребованными как при анализе многомерных временных рядов (MSSA, Multivariate or Multi-channel SSA), так как для анализа цифро-

вых изображений (2D-SSA). Методы исходно не были привязаны к SSA, в частности, метод MSSA назывался методом EEOF (extended empirical orthogonal functions).

Еще одной целью данной работы является построение общей схемы метода и его реализации, позволяющей применять метод к объектам любой размерности, от многомерных временных рядов до n -мерных изображений. При этом как сам объект, так и скользящее по нему окно могут иметь произвольную форму. Общий подход позволяет обобщить результаты для одномерного SSA на общий случай.

Любой метод анализа реальных данных оказывается нерабочим без его эффективной реализации. Размер временных рядов и, тем более, изображений, может быть очень большим, поэтому вычислительные затраты времени и памяти должны быть разумными. Исходно существовало мнение, что метод очень трудоемок и поэтому не применим к данным большого объема. Существующие ранее реализации метода были сделаны непосредственно следя алгоритму, без поиска эффективной реализации. Поэтому весьма актуальным является разработка пакета, который, во-первых, используется быстрые реализации математических процедур, а, во-вторых, позволяет единообразно анализировать объекты разной формы и размерности.

Актуальность темы

Существует много подходов к временным рядам, часть из которых решает узкие задачи или подходит для конкретных моделей данных, а часть претендует на решение целого круга задач. Основной задачей является прогноз ряда и многие методы разрабатываются именно для прогноза. К таким методам можно отнести, например, методы типа ARIMA или методы типа ETS (модели с экспоненциальным сглаживанием). Эти методы умеют работать с данными, которые включают в себя тренд и периодичность, однако, например, значение периода должно быть задано и он должен быть один. К тому же, предъявляются довольно сильные требования к числу периодов, которое должно помещаться в длину ряда. Для ETS вид тренда должен быть задан. Прогноз в ARIMA сильно зависит от количества дифференцирований ряда. Большим преимуществом этих методов является то, что происходит автоматический подбор параметров, правда, в определенной модели данных. Таким образом, классические методы обладают как достоинствами, так и недостатками.

Методы типа SSA также умеют решать очень широкий круг задач для временных рядов разной структуры. Одним из главных преимуществ метода является то, что не требуется задавать исходно модель временного ряда (в частности, величину периода и форму тренда); метод часто называют непараметрическим, что как является большим достоинством метода SSA, так и сопровождается недостатком в виду сложности автоматизации метода в случае его непараметрического применения.

Применению метода SSA посвящено огромное множество разрозненных работ, решающих разные задачи анализа и прогноза временных рядов, в той или иной степени опирающихся на теорию метода SSA. Поэтому выработка общего подхода к теории и применению

SSA является важной задачей.

Цели диссертационной работы

Основными целями диссертационной работы являются следующие.

1. Создание общей методологии метода SSA.
2. Единый подход к развитию и структуре методов из семейства SSA.
3. Улучшение SSA-разделимости интерпретируемых компонент изучаемых объектов (временных рядов, цифровых изображений) для увеличения точности их выделения.
4. Общий подход к многомерным обобщениям SSA, позволяющим обрабатывать объекты различной размерности и формы.
5. Выбор параметров метода, проверка гипотезы о существовании сигнала.
6. Методологическая поддержка R-пакета RSSA с точки зрения его структуры и реализации.
7. Исследование возможностей метода в применении к реальным задачам.

Методология и методы исследования

Методологией исследования, проведенного в диссертации, является комплексный подход к проблеме, позволяющий масштабировать предлагаемые алгоритмы на объекты разной структуры. В качестве методов при этом использовались как теоретические методы с применением линейной алгебры, математического анализа, теории вероятностей и математической статистики, так и методы численного исследования, основанные на статистическом моделировании. Для реализации алгоритмов использовался язык программирования R.

Основные положения, выносимые на защиту

1. Создана общая схема методов SSA, позволяющая как строить различные разложения временного ряда, адаптивно подстраиваясь к его структуре, так и включать в себя расширения, позволяющие анализировать разные объекты, такие как временные ряды, системы временных рядов, цифровые изображения и многомерные объекты различной формы.
2. Предложены и обоснованы алгоритмы ProjSSA, Iterative OSSA, DerivSSA, SSA-ICA, позволяющие улучшить качество выделения компонент изучаемых объектов, таких как тренд/паттерн, регулярные колебания и шум.

3. Задача оценки сигнала методом наименьших квадратов рассмотрена как задача взвешенной аппроксимации ганкелевыми матрицами малого ранга. Доказана невозможность выбора весов для достижения эквивалентности этих задач.
4. Исследовано влияние параметров метода SSA на качество выделения компонент временного ряда. Предложены методы автоматической идентификации компонент временного ряда.
5. Разработан общий подход к многомерным обобщениям SSA, называемый ShapedSSA, позволяющий в едином стиле обрабатывать не только объекты прямоугольной формы, но и объекты сложной формы, начиная от временных рядов с пропусками и заканчивая многомерными объектами сложной формы.
6. Для проверки гипотезы о (не)существовании сигнала во временном ряде предложен статистический тест, контролирующий групповую ошибку.
7. Разработан методологический подход к структуре и реализации пакета RSSA, что позволило сделать эффективную реализацию с удобным интерфейсом.
8. Метод SSA был применен к задачам предсказания параметров вращения Земли, оценке плотностей распределений, построению параметрической модели профилей экспрессии гена бикоид, выделению паттернов в многомерных данных экспрессии генов.

Научная новизна

Основу научной новизны составляет целостный подход ко всему многообразию методов и исследуемых объектов. Полученные теоретические результаты являются новыми.

Теоретическая и практическая значимость

Теоретическая ценность состоит в теоретическом обосновании алгоритмов и их свойств, что позволяет получать более точные результаты анализа.

Практическая ценность состоит в применении построенных алгоритмов к реальным данным и расширении круга практических задач, к которым могут быть применены методы семейства SSA. Она подтверждается возможностью получить новые результаты в той области, к реальным задачам которой применяется один из SSA методов.

Апробация работы

Доклады Основные результаты докладывались на следующих семинарах и конференциях.

1. Научный семинар кафедры статистического моделирования.

2. Всероссийская научная конференция СПИСОК, 2016, 2017, 2019, 2022.
3. Научный семинар в School of Mathematics, Кардиффский университет, Великобритания, 2019.
4. Приглашенный доклад на Colloquium in School of Mathematics, Кардиффский университет, Великобритания, 2017.
5. Приглашенный доклад на международной конференции ‘Structured low-rank approximation’, Гренобль, Франция, 2015.
6. Приглашенный доклад на международной конференции ‘Optimal decisions in statistics and data analysis’, Кардифф, Великобритания, 2013.
7. Приглашенный доклад на международной конференции ‘Singular Spectrum Analysis and its applications’, Пекин, Китай (2012).

Гранты

1. РНФ 23-21-00222. Интеллектуальный анализ временных рядов с помощью методов, основанных на подпространстве сигнала (2023–2024, руководитель)
2. РФФИ 20-01-00067. Разработка математических методов анализа и прогнозирования одномерных и многомерных временных рядов в рамках анализа сингулярного спектра (2020–2022, руководитель)
3. РФФИ 16-04-00821. Анализ, классификация и моделирование паттернов экспрессии генов в раннем эмбриогенезе ряда объектов биологии развития (2016–2018, руководитель)
4. РФФИ 15-04-06480. Молекулярные механизмы закладки осевой организации зачатков в эмбриогенезе на примере модельных насекомых. (2015–2017, исполнитель)
5. РФФИ 13-04-02137. Вариабельная экспрессивность генов сегментации дрозофилы и механизмы фенотипической стабильности. (2013–2014, исполнитель)

Публикации

Статьи

По теме диссертации опубликовано 22 работы в журналах, индексируемых в Scopus/WoS: [4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25].

Сборники трудов конференций и главы в книгах

По теме диссертации опубликовано 3 работы в сборниках трудов конференций, индексируемых в Scopus/WoS: [26, 27, 28].

Монографии

По теме диссертации написаны 3 монографии (в соавторстве) [1, 29, 30], одна из которых вышла также во втором расширенном издании: [31].

Остальные публикации

В сборниках и трудах конференций вышли публикации [32, 33, 34, 35, 36, 37, 38, 39, 40, 41].

В журналах дополнительно опубликованы статьи [42, 43, 44, 45].

В диссертацию включены результаты из статей, полученные автором или в большой степени автором при совместной работе над результатом.

Основное содержание по главам

Диссертационная работа состоит из введения, семи глав, заключения и библиографии. Ниже коротко опишем содержание глав и приведем работы, в которых содержатся основные результаты.

В главе 1 предлагается общая схема методов семейства SSA, в терминах которой описываются основные понятия метода SSA и структура подачи материала в следующих главах. Даётся схема метода, в которую вкладываются как модификации метода, улучшающие разделимость компонент временного ряда, так и многомерные обобщения метода. Первые шесть разделов главы являются обзором методов семейства SSA. Два последних раздела описывают базовые методы SSA анализа и прогноза временных рядов. Обзор основан на работе [20].

В главах 2 и 3 рассматриваются задачи обработки одномерных временных рядов. В главе 2 предлагаются различные методы улучшения разделимости путем изменения шага разложения. Эти методы описаны в статьях: SSA с проекцией [16], Iterative Oblique SSA [9], Filter-adjusted O-SSA и SSA с производными [9], SSA-ICA [14]. Глава 2 завершается разделом, описывающим методы автоматической группировки для выделения трендовых и периодических компонент, для применимости которых компоненты сигнала должны быть разделимы, например, с помощью методов, описанных в предыдущих разделах.

Глава 3 посвящена решению некоторых задач, связанных с моделью, которой компонента ряда точно или приближенно удовлетворяет. Первая задача — заполнение пропусков во временном ряде [4].

Вторая задача — оценка сигнала, который является рядом конечного ранга. Алгоритмы для решения этой задачи предложены в работах [15, 22, 24]. В [13, 19] обсуждаются соотношения весов в двух разных постановках задачи для их эквивалентности.

Третья задача — обнаружение сигнала в зашумленном временном ряде методом Монте-Карло SSA. В качестве шума рассматривается красный шум (это специфика метода SSA), в качестве сигнала — синусоидальный сигнал с неизвестной частотой. В качестве решения задачи строится критерий проверки гипотезы о том, что ряд состоит только из шума. Название метода не совсем отражает суть, так как здесь не строится разложение временного ряда методом SSA, а только используются сингулярные вектора траекторной матрицы для постро-

ения критерия. “Монте-Карло” в названии означает, что распределение статистики критерия строится с помощью моделирования суррогатных данных. Хотя метод Монте-Карло SSA был предложен уже давно, однако он не рассматривался с точки зрения множественного тестирования, что сделано в работе [25].

Завершает главу 3 обсуждение выбора параметров в методе SSA для решения задач анализа и прогноза [5].

Хотя расширение MSSA метода SSA для анализа системы временных рядов можно рассматривать как частный случай Shaped SSA, который рассматривается в главе 5, методу MSSA посвящена отдельная глава 4. Кроме описания метода MSSA в главу 4 включено описание разложения системы временных рядов и его свойств, обсуждение особенностей метода MSSA, введено понятие согласованных рядов и численно сравнивается MSSA, примененный к системе из двух рядов, и SSA, примененный к каждому ряду по-отдельности. Кроме этого, приведено обоснование быстрого векторного SSA и MSSA прогноза. Основные результаты этой работы есть в [12]. Также, в главе приведены алгоритмы автоматической группировки в MSSA.

Подход к методам семейства SSA как частным случаям Shaped SSA описан в Главе 5 следующей работе [12]. Кроме описания Shaped SSA, глава содержит описание частных случаев, таких как Shaped 1D-SSA, применимый к рядам с пропусками, MSSA и 2D-SSA, теория которого описана в [40].

В короткой главе 6 мы представляем пакет RSSA, в котором методы семейства RSSA реализованы эффективно и в едином стиле, описанном в предыдущих главах [8, 12].

Последняя глава 7 содержит ряд приложений метода SSA. Они относятся к прогнозу параметров вращения Земли [28], оценки плотности по эмпирическим данным [7], анализ экспрессии генов в большом количестве работ [17, 21, 10, 11, 26, 27, 18, 6], поэтому последней теме посвящены три раздела.

Используемые обозначения

SVD	сингулярное разложение
ЛРС	линейное рекуррентное соотношение
SSA	анализ сингулярного спектра (singular spectrum analysis)
HSLRA	ганкелева структурная аппроксимация малого ранга (Hankel structured low-rank approximation)
\mathbf{X} или \mathbf{X}_N	временной ряд или упорядоченный набор чисел
N	длина или размер \mathbf{X}
L	длина окна
$K = N - L + 1$	число L -вложенных векторов, полученных из \mathbf{X}
\mathcal{T}	оператор вложения
$\mathbf{X} = \mathcal{T}(\mathbf{X})$	траекторная матрица, связанная с \mathbf{X}
$\ \mathbf{X}\ _F$	норма матрицы Фробениуса
rank \mathbf{X}	rank of \mathbf{X}
$\mathcal{X} = \mathcal{X}^{(L)}(\mathbf{X})$	L -траекторное пространство \mathbf{X}
rank $_L(\mathbf{X})$	L -ранг \mathbf{X}
$\Pi_{\mathcal{H}}$	оператор ганкелизации
λ_i	i -ое собственное значение матрицы \mathbf{XX}^T
U_i	i -ый собственный вектор матрицы \mathbf{XX}^T
$V_i = \mathbf{X}^T U_i / \sqrt{\lambda_i}$	i -ый факторный вектор матрицы \mathbf{X}
$(\sqrt{\lambda_i}, U_i, V_i)$	i -я собственная тройка SVD матрицы \mathbf{X}
\mathbf{I}_M	единичная $M \times M$ матрица
\mathbb{R}^L	евклидово пространство размерности L
$\mathbb{R}^{L \times K}$	пространство $L \times K$ матриц
span(P_1, \dots, P_n)	линейная оболочка векторов P_1, \dots, P_n

Глава 1

Общая структура методов SSA-семейства

В этой главе мы описываем, как методология SSA может быть расширена в различных направлениях, а также структурируем различные подходы и модификации в рамках SSA. Кроме того, вводим основные понятия, принципы и идеи SSA, представляем единый взгляд на SSA.

Мы начинаем раздел 1.1 с краткого описания решенных задач.

Раздел 1.2 описывает общую структуру всех методов семейства SSA и вводит основные понятия, необходимые для понимания различных версий SSA и эффективного применения SSA на практике.

Раздел 1.3 структурирует различные версии SSA. Как объясняется в этом разделе, существует два взаимодополняющих направления, в которых могут быть созданы версии SSA: одно связано с геометрическими особенностями объекта X , а другое определяется выбором процедуры разложения траекторной матрицы на матрицы единичного ранга. Эти два направления вариаций SSA не связаны друг с другом, так что любое расширение SSA, связанное с геометрией X может быть совмещено с любой процедурой разложения траекторной матрицы.

В разделе 1.4 обсуждается понятие разделимости, которое является наиболее теоретически важным понятием SSA. Достижение разделимости (например, сигнала от шума) является ключевой задачей SSA в большинстве приложений. Поэтому правильное понимание этой концепции необходимо для того, чтобы сделать конкретное применение SSA надежным и эффективным. Мы будем возвращаться к разделимости во многих обсуждениях в этой работе.

В разделе 1.5 кратко представлена основная базовая модель, используемая для применения SSA для таких распространенных задач, как прогнозирование, заполнение недостающих данных и мониторинг структурной устойчивости временных рядов. В одномерном случае эта модель предполагает, что часть ряда может быть описана линейной рекуррентным соотношением и, в частности, суммой затухающих синусоид. Оценка параметров этой модели часто является основной задачей при обработке сигналов.

В разделе 1.6 мы кратко обсуждаем место SSA среди других методов анализа временных рядов, обработки сигналов и изображений, а также в разделе представлен краткий обзор недавних публикаций, в которых было проведено сравнение SSA с несколькими традиционными методами.

В разделе 1.7 мы сделаем краткий исторический обзор SSA, обратимся к недавним приложениям SSA и к работам, в которых обсуждается сочетание SSA с другими методами.

Поскольку главы 2–5 содержат в основном оригинальные результаты, мы добавляем к этой главе разделы 1.8 и 1.9 с известными алгоритмами, которые будут использоваться в дальнейшем.

Материал этой главы содержится в работах [12] and [20]. Материал разделов 1.8 и 1.9 изложен, следуя книгам [1, 31].

1.1. Временные ряды и цифровые изображения: общие проблемы

Разложение

Пусть мы наблюдаем временной ряд $X = (x_1, \dots, x_N)$, где $X = T + P + N$, T — тренд, P содержит регулярные колебания, а N — шум. Общая проблема состоит в том, чтобы построить разложение $X = \tilde{T} + \tilde{P} + \tilde{N}$, см. рис. 1.1, позволяющее оценить тенденцию T , весь сигнал $S = T + P$, или периодические компоненты P (P может состоять из периодических компонент с различными фундаментальными периодами). Как только сигнал оценен и его структура определена, можно применять различные методы, основанные на сигнале, такие как прогнозирование.

Если временнй ряд не содержит сигнала, то можно поставить задачу фильтрации; эта задача снова сводится к построению разложения, на низкочастотные и высокочастотные компоненты.

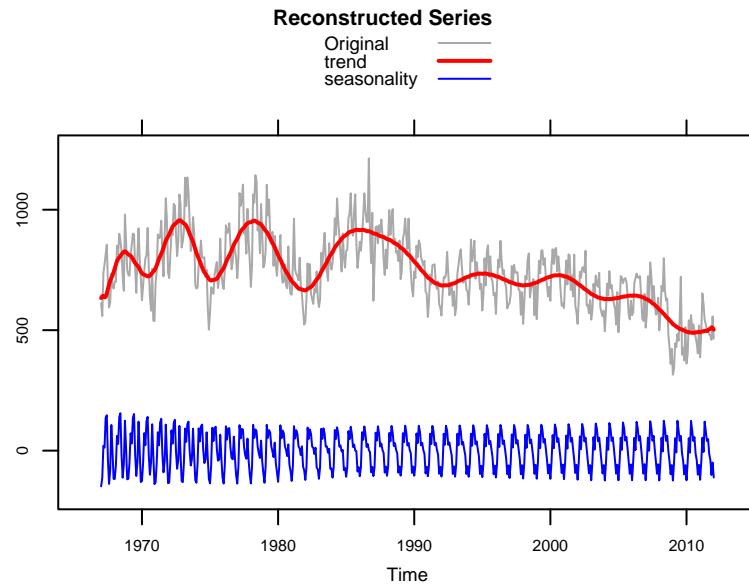


Рис. 1.1: ‘MotorVehicle’, ежемесячные продажи: Разложение

Многие проблемы в обработке цифровых изображений похожи на проблемы, указанные для временных рядов. Пусть

$$X = \begin{pmatrix} x_{1,1} & \dots & x_{1,N_2} \\ \dots & \dots & \dots \\ x_{N_1,1} & \dots & x_{N_1,N_2} \end{pmatrix}$$

цифровое изображение, которое моделируется в виде разложения $X = T + P + N$ на узор, регулярные колебания (например, текстуру) и шум. Тогда снова возникает проблема оценки

компонентов разложения. См. рис. 1.2, который демонстрирует извлечение паттерна (который получается после удаления регулярных колебаний).

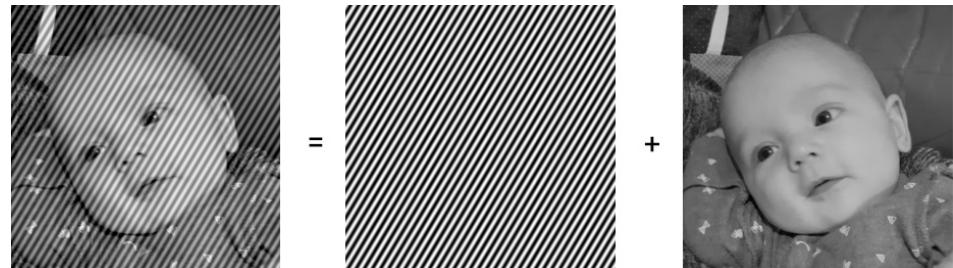


Рис. 1.2: ‘Майя’: Разложение изображения.

Обратим внимание, что ‘цифровые изображения’ — это общее название для двумерных данных, поскольку одно из измерений может быть временным. Проблема разложения для многомерных данных важна и для данных большей размерности. Эти размерности снова могут иметь различную природу. Например, можно рассматривать как трехмерные пространственные данные, так и двухмерные данные с третьим времененным измерением как трехмерные данные; и так далее. Мы будем называть данные размерности n как n D данными.

Другие проблемы временных рядов/изображений

Результаты разложения (в частности, выделения сигнала) позволяют решать множество задач для объектов с различными размерами и формами; среди этих проблем выделим следующие:

- выделение тренда/тенденции;
- сглаживание/фильтрация;
- подавление шума;
- извлечение периодических компонентов (включая сезонную корректировку);
- оценка частоты;
- построение модели и оценка параметров;
- прогнозирование/предсказание (извлеченного сигнала);
- заполнение недостающих данных;
- обнаружение точки разладки.

Таким образом, разложение служит отправной точкой для решения многих важных задач.

1.2. Общая схема семейства SSA и основные понятия

Мы используем SSA как общий термин для семейства методов, основанных на последовательном применении четырех шагов, схематически представленных на рис. 1.3 ниже и кратко описанных в следующем разделе.

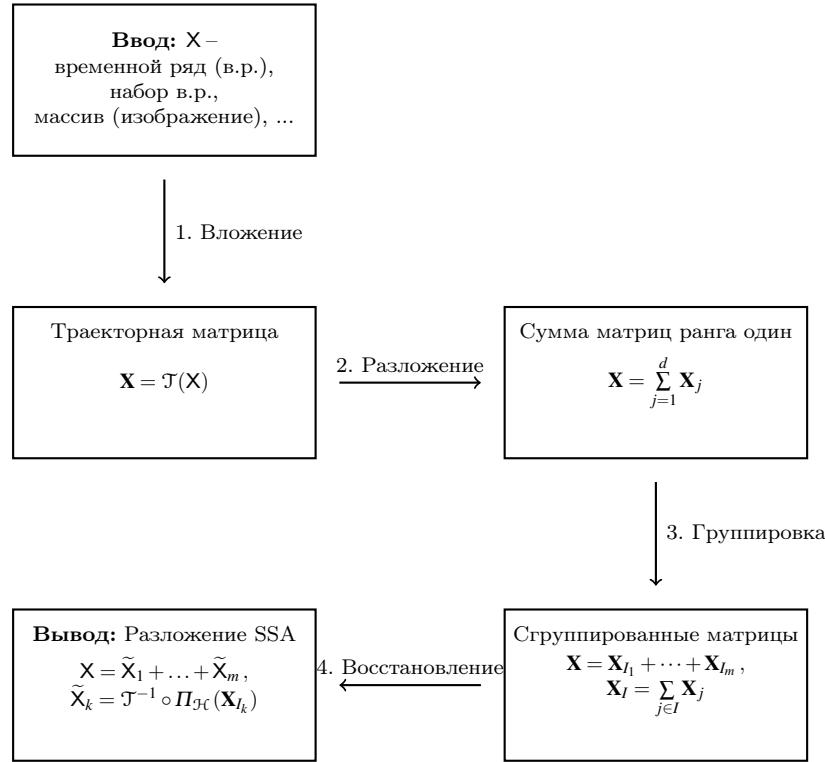


Рис. 1.3: Семейство SSA: Общая схема.

1.2.1. Общий метод SSA

Мы определяем метод SSA (или просто SSA) как любой метод, выполняющий четыре шага, изображенные на рис. 1.3 и кратко описанные ниже. Входной объект X — это упорядоченный набор из N чисел (например, временной ряд или цифровое изображение). Мы обозначаем множество таких объектов через \mathcal{M} . Если не указано иное, значения в X предполагаются вещественными числами, хотя имеется прямое обобщение основного метода SSA на случай комплексных чисел.

Вход: X , упорядоченный набор из N чисел.

Выход: Разложение X на сумму идентифицируемых компонентов: $X = \tilde{X}_1 + \dots + \tilde{X}_m$.

Шаг 1. Вложение. Так называемая *траекторная матрица* $X = \mathcal{T}(X)$, где \mathcal{T} — линейное отображение, преобразующее объект X в $L \times K$ матрицу определенной структуры. Обозначим

множество всех возможных траекторных матриц через $\mathcal{M}_{L,K}^{(\text{H})}$. Буква Н используется для того, чтобы подчеркнуть, что эти матрицы имеют структуру, связанную с ганкелевой.

В качестве примера, в 1D-SSA (т.е. SSA для анализа одномерных вещественных временных рядов), пусть $\mathbf{X} = (x_1, \dots, x_N)$ и $\mathcal{T} = \mathcal{T}_{\text{1D-SSA}}$ отображает \mathbb{R}^N в пространство ганкелевых матриц $L \times K$ с равными значениями на антидиагоналях, где N — длина ряда, L — длина окна, которая является параметром, и $K = N - L + 1$:

$$\mathcal{T}_{\text{1D-SSA}}(\mathbf{X}) = \begin{pmatrix} x_1 & x_2 & x_3 & \dots & x_K \\ x_2 & x_3 & x_4 & \dots & x_{K+1} \\ x_3 & x_4 & x_5 & \dots & x_{K+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_L & x_{L+1} & x_{L+2} & \dots & x_N \end{pmatrix}. \quad (1.1)$$

Шаг 2. Разложение \mathbf{X} в сумму матриц ранга 1. Результатом этого шага является разложение

$$\mathbf{X} = \sum_i \mathbf{X}_i, \quad \mathbf{X}_i = \sigma_i U_i V_i^T, \quad (1.2)$$

где $U_i \in \mathbb{R}^L$ и $V_i \in \mathbb{R}^K$ — векторы, такие что $\|U_i\| = 1$ и $\|V_i\| = 1$ для всех i и σ_i — неотрицательные числа.

Основным примером такого разложения является обычное сингулярное разложение (singular value decomposition, SVD) для вещественных матриц \mathbf{X} . Если используется обычное SVD, то соответствующий метод SSA мы называем ‘Basic SSA’ [1, Глава 1]. Пусть $\mathbf{S} = \mathbf{X}\mathbf{X}^T$, $\lambda_1 \geq \dots \geq \lambda_L \geq 0$ — собственные значения матрицы \mathbf{S} , $d = \text{rank } \mathbf{X} = \max\{j : \lambda_j > 0\}$, U_1, \dots, U_d — соответствующие собственные векторы, и $V_j = \mathbf{X}^T U_j / \sqrt{\lambda_j}$, $j = 1, \dots, d$, являются факторными векторами. Обозначим $\mathbf{X}_j = \sqrt{\lambda_j} U_j V_j^T$. Тогда SVD траекторные матрицы \mathbf{X} можно записать как

$$\mathbf{X} = \mathbf{X}_1 + \dots + \mathbf{X}_d. \quad (1.3)$$

Тройка $(\sqrt{\lambda_j}, U_j, V_j)$, состоящая из сингулярного числа $\sigma_i = \sqrt{\lambda_j}$, левого сингулярного вектора U_j и правого сингулярного вектора V_j \mathbf{X} называется *jой собственной тройкой*.

Шаг 3. Группировка. Входными данными на этом этапе являются разложение (1.2) и спецификация того, как группировать компоненты (1.2).

Пусть $I = \{i_1, \dots, i_p\} \subset \{1, \dots, d\}$ — множество индексов. Тогда результирующая матрица \mathbf{X}_I , соответствующая группе I определяется как $\mathbf{X}_I = \mathbf{X}_{i_1} + \dots + \mathbf{X}_{i_p}$.

Предположим, что разбиение множества индексов $\{1, \dots, d\}$ на m непересекающихся подмножеств I_1, \dots, I_m . Тогда результатом этапа группировки является *группированное раз-*

разложение матрицы

$$\mathbf{X} = \mathbf{X}_{I_1} + \dots + \mathbf{X}_{I_m}. \quad (1.4)$$

Если указано только одно подмножество I из $\{1, \dots, d\}$, то все равно можно считать, что дано разбиение $\{1, \dots, d\}$: это разбиение состоящее из двух подмножеств, I и $\bar{I} = \{1, \dots, d\} \setminus I$. В этом случае \mathbf{X}_I обычно ассоциируется с интересующим нас паттерном (например, сигналом) а $\mathbf{X}_{\bar{I}} = \mathbf{X} - \mathbf{X}_I$ может рассматриваться просто как остаток.

Группировка разложения (1.2), где $I_k = \{k\}$, называется *элементарной*.

Шаг 4. Восстановление. На этом шаге каждая матрица \mathbf{X}_{I_k} из разложения (1.4) переводится обратно в форму входного объекта \mathbf{X} . Это преобразование выполняется оптимально в следующем смысле: для матрицы $\mathbf{Y} \in \mathbb{R}^{L \times K}$ мы ищем объект $\tilde{\mathbf{Y}} \in \mathcal{M}$, который обеспечивает минимум для $\|\mathbf{Y} - \mathcal{T}(\tilde{\mathbf{Y}})\|_F$, где $\|\mathbf{Z}\|_F = (\sum_{ij} |z_{ij}|^2)^{1/2}$ — это норма Фробениуса для $\mathbf{Z} = [z_{ij}] \in \mathbb{R}^{L \times K}$.

Пусть $\Pi_{\mathcal{H}} : \mathbb{R}^{L \times K} \rightarrow \mathcal{M}_{L,K}^{(H)}$ является ортогональной проекцией из $\mathbb{R}^{L \times K}$ на $\mathcal{M}_{L,K}^{(H)}$ по норме Фробениуса. Тогда $\tilde{\mathbf{Y}} = \mathcal{T}^{-1} \circ \Pi_{\mathcal{H}}(\mathbf{Y})$. Проекция $\Pi_{\mathcal{H}}$ — это просто усреднение значений, соответствующих данному элементу объекта, подробности см. в разделе 1.2.2. Например, в 1D-SSA композиция отображений $\mathcal{T}^{-1} \circ \Pi_{\mathcal{H}}$ использует усреднение по антидиагоналям, так что $\tilde{y}_k = \sum_{(i,j) \in \mathcal{A}_k} (\mathbf{Y})_{ij} / |\mathcal{A}_k|$, где $\mathcal{A}_k = \{(i,j) : i+j = k+1, 1 \leq i \leq L, 1 \leq j \leq K\}$.

Пусть $\hat{\mathbf{X}}_k = \mathbf{X}_{I_k}$ — восстановленные матрицы, $\tilde{\mathbf{X}}_k = \Pi_{\mathcal{H}} \hat{\mathbf{X}}_k$ — соответствующие траекторные матрицы и $\tilde{\mathbf{X}}_k = \mathcal{T}^{-1}(\tilde{\mathbf{X}}_k)$ — соответствующие восстановленные объекты. Тогда полученное разложение исходного объекта \mathbf{X} является

$$\mathbf{X} = \tilde{\mathbf{X}}_1 + \dots + \tilde{\mathbf{X}}_m. \quad (1.5)$$

Если группировка элементарная, то восстановленные объекты $\tilde{\mathbf{X}}_k$ в (1.5) называются *элементарными компонентами*.

Для удобства ссылок, этапы 1 и 2 общей схемы SSA иногда объединяют в так называемый ‘этап разложения’, а этапы 3 и 4 — в ‘этап восстановления’.

1.2.2. Основные понятия

Параметры методов SSA

Шаг 1: параметры линейного отображения \mathcal{T} . Для заданного объекта \mathbf{X} траекторная матрица $\mathbf{X} = \mathcal{T}(\mathbf{X})$ может быть вычислена различными способами. В 1D-SSA есть только один параметр на шаге 1, длина окна L .

Шаг 2: нет параметров, если выполняется обычное SVD. В противном случае, если используется альтернативное разложение \mathbf{X} в сумму матриц ранга один, может потребоваться несколько параметров, см. раздел 1.3.1.

Шаг 3: параметр (или параметры), определяющий группировку.

Шаг 4: никаких дополнительных параметров.

Разделимость

Очень важным понятием в методологии SSA является разделимость. Пусть $\mathbf{X} = \mathbf{X}_1 + \mathbf{X}_2$. (Приближенная) разделимость \mathbf{X}_1 и \mathbf{X}_2 означает, что существует группировка такая, что восстановленный объект $\tilde{\mathbf{X}}_1$ (примерно) равен \mathbf{X}_1 . Представление $\mathbf{X} = \mathbf{X}_1 + \mathbf{X}_2$ может быть связано с множеством различных моделей, таких как “сигнал плюс шум”, “тренд плюс остаток” и “текстура плюс основное изображение”.

Если $\mathbf{X} = \mathbf{X}_1 + \mathbf{X}_2$ и \mathbf{X}_1 и \mathbf{X}_2 приближенно разделимы, то SSA может потенциально отдельить \mathbf{X}_1 от \mathbf{X}_2 ; т.е., он может найти разложение $\mathbf{X} = \tilde{\mathbf{X}}_1 + \tilde{\mathbf{X}}_2$ так, что $\tilde{\mathbf{X}}_1 \approx \mathbf{X}_1$ и $\tilde{\mathbf{X}}_2 \approx \mathbf{X}_2$.

Рассмотрим в качестве примера Basic SSA. Свойства SVD дают нам, что (приближенная) ортогональность столбцов и строк траекторных матриц \mathbf{X}_1 и \mathbf{X}_2 от \mathbf{X}_1 и \mathbf{X}_2 можно рассматривать как естественные условия (приближенной) разделимости.

Существует хорошо разработанная теория разделимости для одномерного временного ряда [1, Разделы 1.5 и 6.1]. Многие важные проблемы разложения, от подавления шума и сглаживания до выявления тренда, периодичности и сигнала могут быть решены с помощью SSA. Успех 1D-SSA в разделении разделяемыми объектами связан с простотой ганкелевой структуры траекторных матриц и оптимальностью SVD.

Мы вернемся к важной концепции разделимости в разделе 1.4, где мы определим основную характеристику, которая используется в SSA для проверки разделимости.

Информация для группировки

Теория SSA показывает способы, помогающие обнаружить компоненты (σ_i, U_i, V_i) в разложении (1.2), связанные с компонентой объекта с определенными свойствами для выполнения правильной группировки при условии разделимости. Одно из правил состоит в том, что U_i и V_i (собственные векторы и факторные векторы в случае Basic SSA) произведенные компонентом объекта, воспроизводят свойства этого компонента. Например, в Basic 1D-SSA собственные векторы, производимые медленно меняющимися компонентами ряда, являются медленно меняющимися, собственные векторы, произведенные синусоидой, являются синусоидами с одинаковой частотой; и так далее. Эти свойства помогают выполнить группировку путем визуального анализа собственных векторов, а также с помощью некоторых автоматических процедур, см. раздел 2.5.

Траекторные пространства и подпространства сигналов

Пусть \mathbf{X} — траекторная матрица, соответствующая некоторому объекту X . *Столбовое (строковое) траекторное пространство* \mathbf{X} — это линейное подпространство, натянутое на

столбцы (соответственно, строки) матрицы \mathbf{X} . Термин “траекторное пространство” обычно означает “столбцовое траекторное пространство”. Столбцовое траекторное пространство является подпространством \mathbb{R}^L , а строковое траекторное пространство — подпространством \mathbb{R}^K . В общем случае, для реальных данных траекторные пространства совпадают с соответствующими евклидовыми пространствами, поскольку они порождаются зашумленными данными. Однако в модели “сигнал плюс шум”, когда сигнал имеет траекторную матрицу неполного ранга, траекторное пространство сигнала можно назвать подпространством сигнала. Можно рассматривать как столбцовые, так и строковые подпространства сигнала; размерности строковых и столбцовых подпространств совпадают.

Объекты конечного ранга

Класс объектов, которые подходят для SSA — это так называемые объекты конечного ранга. Мы говорим, что объект (либо временной ряд, либо изображение) имеет L -ранг r , если ранг его траекторной матрицы $r < \min(L, K)$; то есть, траекторная матрица матрица является матрицей неполного ранга. Если L -ранг r не зависит от выбора L для любых достаточно больших размеров объекта и траекторной матрицы, то мы говорим, что объект имеет конечный ранг и имеет ранг r , см. раздел 1.8.2 для строгих определений.

Поскольку траекторные матрицы, рассматриваемые в методах SSA, являются либо чисто ганкелевыми, либо состоят из ганкелевых блоков, то траекторные матрицы неполного ранга тесно связаны с объектами, удовлетворяющими некоторым линейным соотношениям. Эти линейные соотношения могут быть использованы для построения методов прогнозирования. В одномерном случае, при некоторых неограничивающих условиях, ганкелевы матрицы неполного ранга находятся в однозначном соответствии с линейными рекуррентными соотношениями (ЛРС) вида

$$x_n = a_1 x_{n-1} + \dots + a_r x_{n-r}$$

и поэтому связаны с временными рядами, которые могут быть выражены как суммы произведений экспонент, многочленов и синусоид, см. раздел 1.8.2.

Каждое расширение SSA порождает класс объектов конечного ранга. Знание рангов объектов конечного ранга может помочь распознать компоненты единичного ранга для восстановления. Например, для восстановления экспоненциального тренда в одномерном случае, нам нужно сгруппировать только один компонент ранга один (экспоненциальная функция имеет ранг 1), в то время как для восстановления синусоиды нам обычно требуется сгруппировать два компонента SVD (ранг синусоиды равен 2).

Реальные временные ряды или изображения, как правило, не имеют конечного ранга. Однако, если данный объект \mathbf{X} является суммой сигнала конечного ранга и шума, то, с учетом приближенной разделимости, SSA может приближенно извлечь сигнал и впоследствии использовать методы, предназначенные для рядов конечного ранга.

Восстановление (усреднение)

Давайте формально опишем операцию восстановления матрицы, используемую в Шаге 4 общей схемы описанной в разделе 1.2.1. По аналогии с одномерным случаем эту операцию можно также назвать “усреднением по диагоналям”, даже если усреднение будет производиться по более сложным схемам.

Предположим, что элементы x_τ объекта $\mathbf{X} = \{x_\tau\}$ индексируются индексом τ , который может быть просто целым положительным числом (для одномерного ряда) или мультииндексом (для цифровых изображений).

Линейное отображение \mathcal{T} задает взаимооднозначное соответствие между \mathcal{M} и $\mathcal{M}_{L,K}^{(H)}$, множеством $L \times K$ матриц заданной структуры. Оно помещает элементы \mathbf{X} на определенные места матрицы $\mathcal{T}(\mathbf{X}) = \mathbf{X} = [(\mathbf{X})_{ij}]$.

Пусть $\mathbf{e}_\tau \in \mathcal{M}$ — объект с 1 в качестве τ -го элемента, а все остальные элементы нулевые. Определим множество индексов

$$\mathcal{A}_\tau = \{(i, j) \text{ таких, что } (\mathbf{E}_\tau)_{ij} = 1\},$$

где \mathbf{E}_τ — матрица

$$\mathbf{E}_\tau = [(\mathbf{E}_\tau)_{ij}] = \mathcal{T}(\mathbf{e}_\tau) \in \mathcal{M}_{L,K}^{(H)}.$$

Если τ — место элемента $x_\tau \in \mathbf{X}$, тогда $(\mathbf{X})_{ij} = x_\tau$ для всех $(i, j) \in \mathcal{A}_\tau$.

Предположим теперь, что $\widehat{\mathbf{X}} \in \mathbb{R}^{L \times K}$ является произвольной $L \times K$ матрицей и нам нужно вычислить

$$\widetilde{\mathbf{X}} = (\widetilde{x}_\tau) = \mathcal{T}^{-1} \circ \Pi_{\mathcal{H}}(\widehat{\mathbf{X}}),$$

сначала сделав ортогональную проекцию $\widehat{\mathbf{X}}$ на множество $\mathcal{M}_{L,K}^{(H)}$ и затем возвращая результат в пространстве объектов \mathcal{M} . Эта операция является расширением операции “диагонального усреднения”, применяемой в 1D-SSA: элементы \widetilde{x}_τ из $\widetilde{\mathbf{X}}$ вычисляются по формуле

$$\widetilde{x}_\tau = \sum_{(i,j) \in \mathcal{A}_\tau} (\widehat{\mathbf{X}})_{ij} / w_\tau = \langle \widehat{\mathbf{X}}, \mathbf{E}_\tau \rangle_F / \|\mathbf{E}_\tau\|_F^2, \quad (1.6)$$

где $\mathbf{E}_\tau = \mathcal{T}(\mathbf{e}_\tau)$, $w_\tau = |\mathcal{A}_\tau| = \|\mathbf{E}_\tau\|_F^2$ это количество элементов в множестве \mathcal{A}_τ , а скалярное произведение Фробениуса $\langle \cdot, \cdot \rangle_F$ определяется как

$$\langle \mathbf{Z}, \mathbf{Y} \rangle_F = \sum_{i,j} z_{ij} y_{ij}. \quad (1.7)$$

1.3. Различные версии SSA

Давайте рассмотрим, как четыре шага общей схемы SSA, сформулированной в разделе 1.2, могут отличаться для различных версий SSA.

Шаг 1: форма объекта X и, следовательно, особенность оператора встраивания \mathcal{T} оказывает большое влияние на то, как выглядит конкретная версия SSA.

Шаг 2: можно использовать не только обычное SVD, но и многие другие разложения X в сумму матриц ранга один.

Шаг 3: формально этот шаг одинаков для всех версий SSA, хотя способы для выполнения группировки могут отличаться.

Шаг 4: оператор вложения \mathcal{T} , определенный на шаге 1, полностью определяет операции, выполняемые на этом шаге.

Таким образом, мы имеем два направления для создания различных версий SSA: первое направление связано с геометрическими особенностями объекта X и формой оператора вложения \mathcal{T} , в то время как второе направление определяется формой разложения на шаге 2. По сути, Шаг 1 определяется формой объекта, поэтому его вариации можно рассматривать как расширения 1D-SSA. Если вместо SVD на шаге 2 используется какое-либо другое разложение X в сумму матриц ранга один, то соответствующий алгоритм мы называем модификацией Basic SSA. Эти два направления вариаций SSA не связаны друг с другом, так что любое расширение Шага 1 может быть объединено с любой модификацией Шага 2.

Мы начнем обсуждение с изложения некоторых модификаций, которые могут быть предложены для использования на Шаге 2.

1.3.1. Разложение X в сумму матриц ранга один

Варианты SSA, связанные с методами разложения

Обычный SVD, сформулированный в описании SSA в разделе 1.2.1, является разложением X в сумму матриц ранга один, которое обладает некоторыми свойствами оптимальности, см. [1, Глава 4]. Поэтому Basic SSA, который представляет собой SSA с обычным SVD, используемым на шаге 2, можно рассматривать как наиболее фундаментальную версию SSA среди всех методов SSA.

Перечислим несколько вариаций SSA, которые могут быть полезны для ответа на различные вопросы в рамках SSA.

Хорошо известной модификацией Basic SSA является *Toeplitz SSA* (Section 1.8.6), который был создан для работы с стационарными временными рядами. Эта модификация разработана для анализа общепринятой оценки автоковариационной матрицы исходного временного ряда и предполагает, что этот временной ряд является стационарным. Однако, если временной ряд X является нестационарным, тогда восстановление, полученное с помощью Toeplitz SSA, может иметь значительную погрешность.

Важной разновидностью SSA является *SSA с проекцией* (Раздел 2.1). Если у нас есть параметрическая модель (которая должна быть линейной по параметрам и согласованной

с предположением о конечном ряде) для одной из компонент ряда, например, тренда одномерного ряда, то сначала выполняется проекция на подходящее подпространство, а затем следует разложение остатка, например, с помощью SVD. Известные методы SSA с однократным центрированием и SSA с двойным центрированием для извлечения постоянных и линейных трендов соответственно являются частными случаями SSA с проекцией. В более общем случае произвольный полиномиальный тренд может быть извлечен с помощью подходящей версии SSA с проекцией. Еще одно применение SSA с проекцией заключается в построении подпространства на основе вспомогательного ряда и проектирования траекторной матрицы основного ряда на это подпространство.

Цель ряда версий SSA состоит в том, чтобы улучшить свойства разделимости SVD. Если мы используем косоугольную версию SVD, то результирующий метод SSA превращается в Oblique SSA. Следующие две версии Oblique SSA оказываются полезными на практике, а именно *Iterative Oblique SSA* (Section 2.2) и *Filter-adjusted Oblique SSA* (Section 2.3). Последний вариант полезен для разделения компонентов с одинаковым вкладом.

Вложенное применение различных версий SSA

Поскольку Oblique SSA не обладает хорошими аппроксимирующими свойствами, он не может заменить Basic SSA, который использует обычный SVD. Поэтому Oblique SSA следует применять вложенным образом, так что сначала используется Basic SSA для извлечения нескольких компонентов без тщательного разделения этих компонентов а затем один из предложенных косоугольных методов используется для разделения смешанных компонентов.

Если мы используем Basic SSA для устранения шума, а затем какую-либо другую версию SSA (например, Independent Component Analysis или Oblique SSA) для улучшения разделимости, то это можно интерпретировать как использование Basic SSA для предварительной обработки и использование другого метода для более тонкого анализа. Однако есть существенная разница между этим подходом и некоторыми методами, упомянутыми в разделе 1.7.2, где Basic SSA используется для препроцессинга, а затем применяется ARIMA или другие методы иной природы. Действительно, когда мы используем Basic SSA для устранения шума и другой метод SSA, например, Oblique SSA для улучшения разделимости, то мы используем подпространство сигнала, оцененное Basic SSA, а не сам оцененный сигнал (напомним, что при переходе от оцененного подпространства сигнала к оцененному сигналу мы получаем дополнительную ошибку).

Схематично продемонстрируем вложенное использование методов следующим образом. Пусть $\mathbf{X} = \mathbf{X}^{(1)} + \mathbf{X}^{(2)} + \mathbf{X}^{(3)}$ — разложение временного ряда и $\mathbf{X} = \tilde{\mathbf{X}}^{(1)} + \tilde{\mathbf{X}}^{(2)} + \tilde{\mathbf{X}}^{(3)}$ — соответствующее разложение траекторной матрицы \mathbf{X} . Пусть на этапе разложения возвращается результат Basic SSA $\mathbf{X} = \tilde{\mathbf{X}}^{(1,2)} + \tilde{\mathbf{X}}^{(3)}$ и предположим, что выбранный вложенный метод производит разложение $\tilde{\mathbf{X}}^{(1,2)} = \tilde{\mathbf{X}}^{(1)} + \tilde{\mathbf{X}}^{(2)}$. Тогда окончательный результат $\mathbf{X} = \tilde{\mathbf{X}}^{(1)} + \tilde{\mathbf{X}}^{(2)} + \tilde{\mathbf{X}}^{(3)}$ и, после усреднения по диагонали, $\mathbf{X} = \tilde{\mathbf{X}}^{(1)} + \tilde{\mathbf{X}}^{(2)} + \tilde{\mathbf{X}}^{(3)}$. При использовании Basic SSA в этой

схеме нет необходимости в восстановлении сигнала, так как для уточненного разложения используется только оцененное подпространство сигнала.

Особенности разложения

Результат шага разложения SSA (Шаг 2) может быть записан в виде (1.2). SVD является частным случаем (1.2) и соответствует ортонормированным системам $\{U_i\}$ и $\{V_i\}$. По аналогии с SVD, мы будем называть (σ_i, U_i, V_i) собственными стройками, σ_i сингулярными числами, U_i левыми сингулярными векторами или собственными векторами, V_i правыми сингулярными векторами, или факторными векторами. Для большинства SSA-разложений, каждый U_i принадлежит пространству столбцов \mathbf{X} в то время как каждый V_i принадлежит пространству строк \mathbf{X} . Такие разложения будем называть *согласованными*.

Если системы $\{U_i\}$ и $\{V_i\}$ линейно независимы, то разложение (1.2) является *минимальным*, то есть имеет наименьшее возможное число слагаемых, равное $r = \text{rank } \mathbf{X}$. Если хотя бы одна из систем $\{U_i\}$ или $\{V_i\}$ не является линейно независимой, то разложение (1.2) не является минимальным. Если разложение (1.2) не является согласованным, то оно может быть не минимальным, даже если $\{U_i\}$ или $\{V_i\}$ линейно независимы, так как их проекции на пространство столбцов (или строк) могут быть зависимыми.

Если обе системы векторов $\{U_i\}$ и $\{V_i\}$ являются ортогональны, то разложение (1.2) называется биортогональным. Если $\{U_i\}$ ортогональны, то разложение называется левоортогональным; если $\{V_i\}$ ортогонально, то разложение называется право-ортогональным.

Если \mathbf{X}_i F-ортогональны так, что $\langle \mathbf{X}_i, \mathbf{X}_j \rangle_F = 0$ для $i \neq j$, то мы говорим, что соответствующие разложения являются F-ортогональны. Для F-ортогональности достаточно либо левой, либо правой ортогональности. Для F-ортогональных разложений (1.2), $\|\mathbf{X}\|_F^2 = \sum_i \|\mathbf{X}_i\|_F^2$. В общем случае, однако, $\|\mathbf{X}\|_F^2$ может отличаться от $\sum_i \|\mathbf{X}_i\|_F^2$.

Вклад k -ой компоненты матрицы \mathbf{X}_k определяется как $\sigma_k^2 / \|\mathbf{X}\|_F^2$, где $\sigma_k^2 = \|\mathbf{X}_k\|_F^2$. Для F-ортогональных разложений сумма вкладов компонентов равна 1. В противном случае эта сумма может значительно отличаться от 1.

Разложения в различных версиях SSA

Соберем несколько версий 1D-SSA, которые реализованы в пакете RSSA и основаны на различных процедурах, используемых на шаге разложения, и укажем их особенности. Некоторые из этих вариантов также реализованы для многомерных версий SSA.

Basic SSA: обычный SVD, согласованное, минимальное, биортогональное и поэтому F-ортогональное разложение.

Toeplitz SSA: в общем случае, несогласованное, неминимальное, F-ортогональное разложение.

SSA с проекцией: F-ортогональное, но несогласованное разложение, если хотя бы один вектор базиса, используемый для проекции, не принадлежит столбцовому (строовому) траекторному пространству. Компоненты, полученные проекциями, располагаются в начале разложения и имеют индексы $1, \dots, n_{\text{special}}$.

Oblique SSA с предварительной обработкой фильтром (Filter-adjusted O-SSA): согласованное, минимальное, F-ортогональное разложение. Основным частным случаем является DerivSSA.

Iterative Oblique SSA (Iterative O-SSA): согласованное, минимальное косоугольное разложение.

Версии разложения Oblique SSA выполняются вложенным образом.

1.3.2. Версии SSA, работающие с различными формами объекта

В этом разделе мы кратко рассмотрим версии SSA, которые оперируют объектами X различных форм. Основное различие между версиями SSA этого раздела является вид оператора вложения \mathcal{T} .

Как было сказано выше, SSA может быть применен к многомерным объектам. SSA для системы рядов называется многомерным (или многоканальным) SSA, сокращенно MSSA (Раздел 4.1); SSA для цифровых изображений в черно-белой шкале называется 2D-SSA (Раздел 5.2.3).

Shaped SSA (Разделы 5.1 и 5.2.1) может обрабатывать данные сложной структуры и произвольной формы; размерность объекта X не имеет значения. Shaped SSA может применяться к различным видам данных, включая временные ряды, системы временных рядов, цифровые изображения прямоугольной и непрямоугольной формы.

Как для рядов, так и для изображений доступны версии Circular SSA. Для рядов Circular SSA работает в метрике круга, и поэтому эта версия подходит для рядов, которые действительно заданы на окружности.

Для изображений Circular SSA дает возможность разложить изображения, заданные на цилиндре (например, полученные как цилиндрическая проекция сферы или эллипсоида) или на торе. Версии Circular SSA позволяют устраниить краевые эффекты, которые неизбежны при например, плоскостной развертке цилиндра.

Таблица 1.1 содержит список расширений, рассматриваемых в этой работе.

Таблица 1.1: Классификация различных версий SSA на основе различных геометрических особенностей объекта X .

Метод	Данные	Обозначение	Траекторная матрица
1D-SSA	временной ряд (в.р.)	$X = (x_1, \dots, x_N)$ длины N	ганселева
MSSA	система в.р.	$X^{(p)}$ длии N_p , $p = 1, \dots, s$	составная ганселева
2D-SSA	прямоугольное изображение	$X = (x_{ij})_{i,j=1}^{N_x, N_y}$	из ганселевых блоков
Shaped SSA	произвольной формы	$X = (x_{(i,j)})_{(i,j) \in \Omega}$	квази-ганселева
Circular SSA	круговой/цилиндрический	объекты на круге/цилиндре	квази-ганселева

1.4. Разделимость в SSA

В этом разделе мы более подробно обсуждаем разделимость SSA; см. также разделы 1.8.3, 2.1.2, 2.2.1, 2.3.1 для специальных случаев.

Предположим, что мы наблюдаем сумму двух объектов $X = X^{(1)} + X^{(2)}$. Мы говорим, что SSA разделяет эти два объекта, если на шаге группировки (шаг 3) можно найти такую группировку, что $\tilde{X}^{(k)} = X^{(k)}$ для $k = 1, 2$. Если эти равенства выполняются приближенно, то это определяет приближенную разделимость. Асимптотическая разделимость может быть введена, если длина ряда $N \rightarrow \infty$. В этом случае приближенная разделимость имеет место для достаточно большой длины ряда. Свойство разделимости очень важно для SSA, так как оно означает, что метод потенциально работает; то есть, он способен извлечь компоненты объекта.

В Basic SSA и его многомерных расширениях, (приближенная) разделимость означает (почти) ортогональность компонентов объекта, так как используется биортогональное SVD-разложение. В других версиях SSA могут быть сформулированы различные условия разделимости.

Если разложение (1.2) на шаге разложения не является единственным, то вводятся два варианта разделимости. Слабая разделимость означает, что существует разложение такое, что после группировки мы получаем $\tilde{X}^{(k)} = X^{(k)}$. Сильная разделимость означает, что это равенство достижимо при любом допустимом разложении.

Условия точной разделимости являются очень жесткими, в то время как асимптотическая разделимость имеет место для широкого диапазона компонент объекта. Например, медленно меняющиеся гладкие компоненты асимптотически отделимы от регулярных колебаний и они обе асимптотически отделимы от шума.

Для того чтобы проверить разделимость восстановленных компонент \tilde{X}_1 и \tilde{X}_2 , мы должны проверить ортогональность их восстановленных траекторных матриц $\tilde{\mathbf{X}}_1$ и $\tilde{\mathbf{X}}_2$. Удобной мерой их ортогональности является скалярное произведение Фробениуса $\langle \tilde{\mathbf{X}}_1, \tilde{\mathbf{X}}_2 \rangle_F$, определенное в (1.7).

Нормированной мерой ортогональности является

$$\rho(\tilde{\mathbf{X}}_1, \tilde{\mathbf{X}}_2) = \langle \tilde{\mathbf{X}}_1, \tilde{\mathbf{X}}_2 \rangle_{\mathbf{F}} / (\|\tilde{\mathbf{X}}_1\|_{\mathbf{F}} \|\tilde{\mathbf{X}}_2\|_{\mathbf{F}}).$$

Поскольку траекторная матрица состоит из $w_{\tau} = |\mathcal{A}_{\tau}| = \|\mathcal{T}(\mathbf{e}_{\tau})\|_{\mathbf{F}}^2$ вхождений τ -го элемента x_{τ} исходного упорядоченного объекта, мы можем ввести взвешенное скалярное произведение в пространстве \mathcal{M} : $(Y, Z)_{\mathbf{w}} = \sum_{\tau} w_{\tau} y_{\tau} z_{\tau}$. Тогда величину

$$\rho_{\mathbf{w}}(\tilde{\mathbf{X}}_1, \tilde{\mathbf{X}}_2) = \rho(\tilde{\mathbf{X}}_1, \tilde{\mathbf{X}}_2) = \frac{(\tilde{\mathbf{X}}_1, \tilde{\mathbf{X}}_2)_{\mathbf{w}}}{\|\tilde{\mathbf{X}}_1\|_{\mathbf{w}} \|\tilde{\mathbf{X}}_2\|_{\mathbf{w}}} \quad (1.8)$$

будет называться **\mathbf{w} -корреляцией** по статистической аналогии. Заметим, однако, что в этом определении средние не вычитаются.

Пусть $\tilde{\mathbf{X}}_j$ — элементарные восстановленные компоненты, полученные в результате элементарной группировки $I_j = \{j\}$. Тогда матрица $\rho_{ij}^{(\mathbf{w})} = \rho_{\mathbf{w}}(\tilde{\mathbf{X}}_i, \tilde{\mathbf{X}}_j)$ называется **\mathbf{w} -корреляционной матрицей**.

Норма $\|\cdot\|_{\mathbf{w}}$ называется *взвешенной нормой* и используется в качестве меры вклада компонентов в разложение (1.5): вклад $\tilde{\mathbf{X}}_k$ определяется как $\|\tilde{\mathbf{X}}_k\|_{\mathbf{w}}^2 / \|\mathbf{X}\|_{\mathbf{w}}^2$.

Если взвешенная корреляция между парой элементарных компонентов велика, то это говорит о том, что эти два компонента сильно коррелируют и, скорее всего, должны быть включены в одну группу.

1.5. Прогнозирование, интерполяция, аппроксимация малого ранга и оценка параметров в SSA

Существует класс объектов, который является специальным для SSA. Это класс объектов, удовлетворяющих линейным рекуррентным соотношениям. Траекторные матрицы этих объектов имеют неполный ранг и, более того, для этих объектов число ненулевых членов в SVD (1.3) не зависит от длины окна, если эта длина достаточно велика. В таких случаях мы будем говорить, что объекты имеют конечный ранг. Класс объектов удовлетворяющих линейным рекуррентным соотношениям, представляет собой естественную модель сигнала для SSA, что является принципиально важным понятием для прогнозирования временных рядов.

Линейное рекуррентное соотношение (ЛРС) для временного ряда $S_N = (s_i)_{i=1}^N$ — это соотношение вида

$$s_{i+t} = \sum_{k=1}^t a_k s_{i+t-k}, \quad 1 \leq i \leq N-t, \quad a_t \neq 0, \quad t < N. \quad (1.9)$$

Хорошо известно (см., например, [46, Теорема 3.1.1]), что последовательность $S_{\infty} = (s_1, \dots, s_n, \dots)$

удовлетворяет ЛРС (1.9) для всех $i \geq 0$ тогда и только тогда, когда для некоторого целого числа p мы имеем для всех n

$$s_n = \sum_{k=1}^p P_k(n) \mu_k^n, \quad (1.10)$$

где $P_k(n)$ — многочлены от n , а μ_k — некоторые комплексные числа. Для вещественных временных рядов (1.10) подразумевает, что класс временных рядов, управляемых ЛРС, состоит из сумм произведений многочленов, экспонент и синусоид.

Упрощенной моделью (1.10) является $s_n = \sum_{k=1}^p C_k \mu_k^n$. Оценка комплексных чисел $\mu_k = \rho_k e^{i2\pi\omega_k}$ эквивалентна оценке частот ω_k и экспоненциального показателя $\ln \rho_k$.

Для изображений $S = [s_{mn}]$, ЛРС являются двумерными и общий член сигнала (который можно назвать паттерном) в рамках модели имеет вид

$$s_{mn} = \sum_{k=1}^p P_k(m, n) \mu_k^m v_k^n. \quad (1.11)$$

Этот важный факт хорошо известен, начиная с [47, §2.20].

Во многих реальных задачах наблюдается зашумленный сигнал (или зашумленный паттерн для изображений) и задача заключается в прогнозировании сигнала, заполнении пропусков в сигнале, оценке параметров сигнала, нахождении точек разладки в сигнале и так далее. Обратим внимание, что не обязательно предполагать, что шум является случайным. В самом общем смысле, шум — это остаток, который не требует дальнейшего исследования.

В рамках SSA возможно получить оценку пространства сигнала, которое является подпространством натянутым на выбранный базис на этапе группировки в SSA. Оценка подпространства сигнала может быть выполнена с помощью итераций (так называемых итераций Кэдзоу), которые заключаются в итеративном применении SSA, см. раздел 1.9.1.

В процессе оценки подпространства сигнала мы также получаем параметрическую оценку сигнала; то есть, оценка элементов множества $\{\mu_k\}$ в одномерном случае и множества $\{\mu_k, v_k\}$ в двумерном случае. Один из наиболее распространенных методов называется ESPRIT для рядов (см. раздел 1.9.2) и его расширение 2D-ESPRIT для изображений [41].

Для временных рядов мы можем прогнозировать найденную структуру, например, путем применения построенного ЛРС. При выполнении интерполяции (которая представляет собой прогнозирование внутрь ряда) можно заполнить пропуски.

1.6. Сравнение SSA с другими методами

В этом разделе мы приводим некоторые замечания и краткий библиографический обзор, касающийся сравнения SSA с несколькими традиционными методами [20].

1.6.1. Преобразование Фурье, фильтрация, шумоподавление

- Преобразование Фурье использует базис, заданный заранее, в то время как SSA использует адаптивный базис, который не ограничивается сеткой частот с разрешением $1/N$. Вейвлет-преобразование также использует фиксированные базисы; преимущество вейвлет-преобразования в том, что изменение частот может быть обнаружено с помощью используемого пространственно-временного базиса. В рамках SSA анализ временных рядов с изменяющейся частотной структурой может быть выполнен с помощью скользящих процедур, например, путем отслеживания подпространства.
- Одним из современных методов оценки частоты является метод высокого разрешения ESPRIT, который является методом, основанным на использовании подпространств. Этот метод можно рассматривать как метод, связанный с SSA, и действительно, он часто применяется как в данной работе, так и в RSSA.
- Преобразование Фурье очень неэффективно для рядов с модуляцией по амплитудам и частотам. SSA может легко справиться с амплитудной модуляцией, но он не может эффективно справиться с частотной модуляцией.
- Разложение SSA иногда можно рассматривать как применение набора линейных фильтров [48, 49, 29] с интерпретацией, зависящей от длины окна L . Для малых L каждая компонента разложения на интервале $[L, K]$, где $K = N - L + 1$, может быть получена с помощью линейного фильтра. Поэтому взгляд с точки зрения фильтрации на результат разложения может быть адекватным. Например, восстановление по ведущим компонентам близко к применению треугольного фильтра.

Если $L \simeq K$ и, следовательно, интервал $[L, K]$ мал, то это не так. В этом случае подход разделимости, основанный на ортогональности разделенных компонентов, является более подходящим. Заметим, что версии Oblique SSA могут ослабить условие ортогональности, см. раздел 2.2.

- Существует большая разница между скользящим средним и SSA для подавления шума. Рассмотрим пример зашумленной синусоиды. Скользящее усреднение добавит погрешность в оценке, вызванное второй производной сигнала, в то время как SSA при больших L обеспечит несмещенную оценку сигнала.

Обратим внимание, что даже при малых L , когда восстановление по ведущей компоненте является взвешенным скользящим средним с положительными весами и поэтому имеет те же недостатки, что и скользящее усреднение, пользователь может добавить дополнительные компоненты для устранения возможного смещения.

- Фильтрация с помощью SSA для снижения уровня шума может быть рассмотрена с точки зрения аппроксимации малого ранга. Хорошие свойства аппроксимации обеспечивают соответствующее подавление шума. Empirical-mode decomposition (EMD), в

свою очередь, начинает построение Intrinsic Mode Functions (IMF) с высоких частот, в то время как тренд содержится в последних IMF.

В качестве примера сравнения SSA, преобразования Фурье и вейвлет-преобразования смотрите, например, [50]. Вывод авторов гласит: “метод фильтрации на основе SSA является надежным для разделения региональных гравитационных аномалий и может быть эффективно использован для фильтрации других геофизических данны”.

В статье [51] метод шумоподавления на основе SSA для удаления помех электрокардиограммы в сигналах электромиографии сравнивается с высокочастотным фильтром Баттервортса, вейвлетами и EMD. Авторы этой работы заявляют: “предложенный подход SSA является эффективным методом удаления артефактов ЭКГ из загрязненных сигналов ЭМГ без использования опорного сигнала ЭКГ”.

В статье [52] сравнивается множество различных методов извлечения тренда для синтетических данных, моделирующих поведение уровня моря; SSA сравнивается со скользящим средним, вейвлетами, регрессией, EMD. Автор пишет: “Оптимальным аналитическим методом, скорее всего, будет SSA, при этом интерактивные методы визуального контроля (ВИ) используются опытными специалистами для оптимизации длины окна и разделимости компонентов”.

Сравнение SSA-фильтрации и фильтров Калмана (KF) можно найти в [53], где показано, что “и SSA, и KF дают многообещающие результаты для станций с сильными сезонными сигналами, в то время как для станций, где доминируют долгосрочные вариации, SSA, похоже, является превосходящим”.

1.6.2. Параметрическая регрессия

Параметрическая регрессия естественным образом предполагает параметрическую модель. Существует большая разница между параметрическими и непараметрическими моделями: если предполагаемая модель верна, то соответствующие параметрические методы являются наиболее подходящими (если в данных нет выбросов). Если предполагаемая параметрическая модель не верна, то результаты параметрических методов являются смещеными и могут вводить в заблуждение. Недостатки непараметрических методов также очевидны: существуют проблемы с прогнозированием, тестированием модели, построением доверительных интервалов и так далее. Часто непараметрические методы служат в качестве инструментов предварительной обработки для параметрических методов. Как обсуждалось в разделе 1.7.2, это часто происходит с SSA.

Сравнение SSA с двойным центрированием и линейной регрессии приведено в разделе 2.1. Оказывается, что SSA с двойным центрированием в качестве метода предварительной обработки значительно повышает точность оценки линейного тренда.

SSA обладает очень редким преимуществом: он может быть непараметрическим методом для предварительного анализа и также может быть параметрическим для моделирова-

ния рядов, управляемых ЛРС. Более того, прогноз с помощью ЛРС использует параметрическую модель в неявном виде; поэтому он более устойчив к отклонениям от модели, чем прогноз, основанный на явной оценке параметров.

Одним из методов построения модели сигнала, управляемого ЛРС, на основе оценки его подпространства является применение HSLRA. HSLRA можно рассматривать как метод оценки параметров в параметрической модели, где задается только ранг сигнала, а не точная параметрическая форма, см. раздел 3.2.

1.6.3. ARIMA и ETS

Во-первых, отметим, что модели (сезонной) ARIMA и экспоненциального сглаживания (ETS, что означает Error, Trend, Seasonal) полностью отличаются от модели SSA (для всестороннего введения в ARIMA и ETS см. [54]). В частности, в ARIMA шум добавляется на каждом рекуррентном шаге, в то время как в SSA шум добавляется после формирования сигнала. Кроме того, тенденции/сезонность в SSA детерминированы, а в ARIMA/ETS тенденции/сезонность случайны. Как и во многих классических методах, ARIMA и ETS требуют задания значений периода для периодических компонентов.

Однако, если рассматривать анализ/прогноз реальных временных рядов, то такие временные ряды не следуют в точности какой-либо модели. Поэтому проблема сравнения методов разной природы не является простой.

Как правило, доверительные интервалы для прогнозов ARIMA слишком велики, но средний прогноз часто может быть адекватным. Преимущество сезонной ARIMA и ETS заключается в том, что модель и ее параметры могут быть подобраны автоматически на основе информационных критериев.

Строго обоснованные информационные критерии еще не построены для SSA. Одной из причин этого является тот факт, что SSA — непараметрический метод. Наиболее стандартный подход к выбору параметров, когда нет заданных моделей, является минимизация ошибки прогнозирования на проверочном периоде. В наиболее часто встречающемся случае, когда прогноз строится на основе r ведущих собственных троек, SSA имеет только два параметра (L и r), которые могут быть оценены путем минимизации ошибки прогнозирования для нескольких прогнозов значений на валидационном периоде.

Сравнение SSA и ARIMA/ETS проводилось во многих работах. Некоторые примеры приведены ниже.

- В [55] показано, что SSA превзошел несколько других методов в примере с прогнозированием прибытия туристов.
- В [56] показано, что для прогнозирования потребностей в скорой помощи “SSA дает превосходные долгосрочные прогнозы (которые особенно полезны для планирования EMS), и сравнимые краткосрочные прогнозы с хорошо известными методами”.

- Автор статьи [57] заключает: “Результаты прогнозирования сравниваются с результатами моделей экспоненциального сглаживания пространства состояний (ETS) и ARIMA. Все три метода одинаково хорошо справляются с процессом прогнозирования. Однако SSA превосходит методы ETS и ARIMA по показателям точности ошибки прогнозирования.”
- В [58, 59], одномерный и многомерный SSA показали хорошие результаты в сравнении с ARIMA и VAR для прогнозирования нескольких рядов европейского промышленного производства.

1.7. Библиографические заметки

Ссылки Истоки сингулярного спектрального анализа (SSA) обычно относят к работам [60] и [61]; хотя алгоритм SSA можно найти, например, в [62]. SSA стал широко известен в климатологии после публикаций [63, 64]. Через несколько лет в книге [65] была собрана основная информация о SSA, существовавшая к тому времени. Параллельно SSA (названный ‘Гусеница’) создавался в России; результаты были опубликованы в [66] (на русском языке). История метода ‘Гусеница’ начинается с [67], где О. М. Калинин указан как автор идей, лежащих в основе метода (трудно найти доступ к этому обзору, поэтому мы ссылаемся на книгу [68, Глава 3, раздел 8], где этот обзор цитируется вместе с алгоритмом первого этапа SSA). Другим источником российской ветви была статья [69].

Большое продвижение в теории SSA было сделано в фундаментальной книге [1], где теория представлена вместе с примерами. Следующая книга — [29] в серии Briefs in Statistics, содержит краткое описание и некоторые обновления с 2001 года, включая описание SSA как набора фильтров. После 2013 года был совершен большой скачок, когда SSA стал методом анализа объектов различных размеров и форм. Кроме того, R-пакет RSSA [70] был разработан с очень быстрой реализацией SSA для различных видов объектов. Предложенный структурированный подход к SSA, его многомерные расширения (MSSA и 2D-SSA) вместе с алгоритмами и описанием реализации в RSSA содержатся в недавней книге [30].

Три упомянутые выше монографии Golyandina с соавторами охватывают очень широкий круг задач, решаемых с помощью SSA, однако в них лишь вскользь обсуждаются практические приложения SSA к стационарным процессам. В то же время, приложения SSA к стационарным временным рядам были разработаны командой из Калифорнийского университета (начиная с [71]), в основном для климатических данных. Некоторые практические приложения, в частности, в экономике и биомедицине, рассматриваются в работах H.Hassani, S.Sanei и их соавторов (см., например, книгу [72] и обзор [73]). Отдельная ветвь связана с реальными задачами в геофизике, где необходимо извлечь следы в виде прямых линий; предварительная обработка производится путем дискретного преобразования Фурье для строк изображения, а затем к коэффициентам Фурье применяется комплекснозначная версия SSA

[74, 75]. Похоже, что эти ветви развиваются несколько независимо друг от друга, поэтому было бы очень полезно им обогатить друг друга.

Схема алгоритма Алгоритм SSA состоит из двух этапов. Первый этап называется разложением, где исследуемый объект (например, временной ряд) преобразуется в траекторную матрицу (ганкелеву матрицу), а затем к траекторной матрице применяется сингулярное разложение для получения разложения на элементарные матричные компоненты ранга один. На втором этапе, который называется восстановление, создаются сгруппированные матричные компоненты и преобразуется разложение на сгруппированные матрицы обратно в разложение исходного объекта путем так называемого диагонального усреднения.

Поскольку идея рассмотрения скользящих отрезков одного временного ряда как различных наблюдений и последующего применения анализа главных компонент (PCA) или преобразования Карунена-Лоэва (KLT) к полученной выборке является простой, этап разложения SSA можно найти во многих работах; трудно определить, какая из них была первой. Например, упомянутые выше работы [67, 60, 61, 63] содержат описание только этапа разложения. Также можно привести [76] и [77] как ссылки, относящиеся к первому этапу SSA. Диагональное усреднение на втором этапе используется в [62, 78, 3, 79, 69]. В настоящее время этап восстановления рассматривается как неотъемлемая часть SSA.

Другое происхождение SSA восходит к свойствам ганкелевых матриц [80]. Иногда происхождение SSA относят к работе [81], где рассматривалось моделирование сигнала в виде суммы экспоненциальных рядов; это происхождение связано с проблемой оценки параметров (частот).

Мы предлагаем называть метод ‘SSA’, если в нем задействованы оба этапа — разложение и восстановление. Методы, основанные только на этапе разложения, называются методами на основе подпространства. Хотя многие методы на основе подпространств были разработаны до SSA, эти методы можно назвать связанными с SSA.

1.7.1. Некоторые недавние применения SSA

Количество публикаций, посвященных применению SSA, неуклонно растет. В дополнение к стандартным областям применения, таким как климатология, метеорология и геофизика, сейчас появилось много работ, посвященных применению в инженерном деле, экономике, финансах, биомедицине и других областях. Можно найти много ссылок на недавние публикации в [82] и много статей в двух специальных выпусках журнала *Statistics and Its Interface* (2010, v.3, No.3 и 2017, v.10, No.1), которые полностью или частично посвящены SSA. В этом разделе мы кратко расскажем о некоторых недавних приложениях SSA. В большинстве этих работ использовались только простейшие версии SSA (то есть Basic SSA из раздела 1.8 и Toeplitz SSA из раздела 1.8.6).

Преимущества 2D-SSA (описанного в разделе 5.2.3) перед некоторыми другими метода-

ми обработки изображений продемонстрированы в работе [83, 84] в применении к гиперспектральной съемке. Применение 2D-SSA для заполнения пропусков рассматривается в статье [85]. В [86] предложено применение многомерного SSA для обнаружения кластеров осцилляторов в многомерных наборах данных.

Сложно найти прикладные области, связанные с анализом временных данных, где не применялся 1D-SSA. Приведем несколько примеров. В [87] и нескольких других работах тех же авторов SSA использовался как основной метод при разработке системы мониторинга износа инструмента. Безопасность мобильных устройств рассматривается в [88], где SSA используется для предварительной обработки. В [89] SSA был использован для извлечения экономических циклов. В [90] SSA используется для заполнения пропусков в данных по осадкам. Некоторые недавние приложения в климатологии были рассмотрены в [91, 92] и в [93]. В [94], SSA помогает решить проблему несанкционированной модификации речевых сигналов. В статье [51], SSA используется для шумоподавления в проблеме удаления помех электрокардиограммы в сигналах электромиографии. Статья [95] связана с разложением и восстановлением многолетних записей цветения восьми видов эвкалиптов. В статье [96] SSA использовался в качестве инструмента предварительной обработки перед классификацией медицинских данных; авторы пишут: “Результаты продемонстрировали надежность подхода при тестировании на крупномасштабных наборах данных с клинически приемлемой чувствительностью и специфичностью”.

1.7.2. SSA для предварительной обработки / комбинация методов

Для многих различных методов SSA обеспечивает улучшение, если он используется в качестве инструмента предварительной обработки. Существуют десятки работ, в которых рассматриваются гибридные методы, включающие SSA. В большинстве приложений SSA служит либо для удаления шума, либо для извлечения признаков. Приведем несколько примеров работ, в которых рассматриваются гибриды SSA и других методов.

SSA используется как препроцессинг для ARIMA в [97]. В работе [98] предлагается совместный гибрид SSA, ARIMA и метода Холта-Уинтерса. В [99] показано, что гибрид SSA + ARMAX лучше, чем только ARMAX для обнаружения структурных повреждений для задач мониторинга состояния конструкций.

В машинном обучении SSA часто используется для получения новых характеристик временных рядов для последующего использования их в других моделях и методах. Это называется извлечением признаков (feature extraction). Статья [100] считается одной из первых, где SSA используется вместе с Support Vector Machines (SVM). Гибрид SSA с нейронными сетями был предложен в работе [101].

В [102], Support Vector Machine Regression (SVR) применяется отдельно к тренду и колебаниям, которые извлекаются с помощью SSA. Построенный метод применяется для прогнозирования временного ряда данных об отказах, собранных на этапе технического обслуж-

живания самолета Boeing 737. Показано, что предложенный гибрид SSA+SVR превосходит методы Холта-Уинтерса, ARIMA, множественной линейной регрессии, групповой метод обработки данных, SSA и SVR, используемые отдельно. Подобные методы рассматриваются в [103], где SSA используется для извлечения тренда и сезонности, а затем нейронные сети и нечеткая логика применяются к ним отдельно с последующим комбинированием. В работе [104] SSA успешно используется для удаления шума перед применением нейронных сетей. Данная работа содержит обзор различных подходов к моделированию дождевого стока с помощью SSA, используемого в комбинации с другими методами.

В [83], SSA был применен в гиперспектральной визуализации для эффективного извлечения признаков (удаления шума), а затем SVM был использован для классификации. Оказалось, что SSA выполняет предварительную обработку лучше, чем Empirical Mode Decomposition (EMD). Обратим внимание, что SSA и EMD не только конкурируют, они могут быть успешными как гибриды. Например, в [105] EMD используется для извлечения тренда а затем SSA применяется для прогнозирования изменений в тренде.

1.8. Basic SSA для одномерных временных рядов

Basic SSA — это SSA для анализа одномерных рядов, в котором разложение на матрицы ранга один на шаге 2 (см. общую схему на рис. 1.3) выполняется с помощью SVD.

1.8.1. Метод

Шаг 1: Вложение

Ряд \mathbf{X} трансформируется в последовательность L -векторов вложения, которые образуют траекторную матрицу $\mathbf{X} = \mathcal{T}_{\text{SSA}}(\mathbf{X})$, как показано в (1.1).

Шаг 2: Разложение

Зададим $\mathbf{S} = \mathbf{XX}^T$ и обозначим через $\lambda_1, \dots, \lambda_d$ положительные *собственные значения* \mathbf{S} , взятые в порядке убывания величины ($\lambda_1 \geq \dots \geq \lambda_d > 0$), и через U_1, \dots, U_d ортонормированную систему *собственных векторов* матрицы \mathbf{S} , соответствующих этим собственным значениям; $V_i = \mathbf{X}^T U_i / \sqrt{\lambda_i}$ называются *факторными векторами*. На этом шаге мы выполняем сингулярное разложение (SVD) траекторной матрицы:

$$\mathbf{X} = \sum_{i=1}^d \sqrt{\lambda_i} U_i V_i^T = \mathbf{X}_1 + \dots + \mathbf{X}_d. \quad (1.12)$$

Матрицы $\mathbf{X}_i = \sqrt{\lambda_i} U_i V_i^T$ в (1.12) имеют ранг 1; такие матрицы называются *элементарными матрицами*. Набор $(\sqrt{\lambda_i}, U_i, V_i)$, состоящий из сингулярного значения $\sqrt{\lambda_i}$, левого сингулярного вектора U_i и правого сингулярного вектора V_i , будет называться *i-ой собственной тройкой*

(сокращенно ЕТ). Заметим, что $\lambda_i = \|\mathbf{X}_i\|_{\text{F}}^2$ и $\|\mathbf{X}\|_{\text{F}}^2 = \|\mathbf{X}_1\|_{\text{F}}^2 + \dots + \|\mathbf{X}_d\|_{\text{F}}^2$. Вклад i -ой компоненты \mathbf{X}_i может быть измерен как $\lambda_i / \sum_j \lambda_j$.

Для реальных временных рядов $d = \text{rank } \mathbf{X}$ обычно равно $\min(L, K)$; то есть, траекторная матрица имеет полный ранг.

Шаг 3: группировка собственных троек

Входными данными на этом шаге являются разложение (1.12) и способ группировки компонент этого разложения.

Пусть $I = \{i_1, \dots, i_p\} \subset \{1, \dots, d\}$ — множество индексов. Тогда результирующая матрица \mathbf{X}_I , соответствующая группе I , определяется как $\mathbf{X}_I = \mathbf{X}_{i_1} + \dots + \mathbf{X}_{i_p}$.

Возьмем разбиение множества индексов $\{1, \dots, d\}$ на m непересекающихся подмножеств I_1, \dots, I_m . Тогда разложение (1.12) приводит к сгруппированному разложению

$$\mathbf{X} = \mathbf{X}_{I_1} + \dots + \mathbf{X}_{I_m}. \quad (1.13)$$

Процедура выбора множеств I_1, \dots, I_m называется *группировкой собственных троек*. Если $m = d$ и $I_j = \{j\}$, $j = 1, \dots, d$, то соответствующая группировка называется *элементарной*.

Группировка осуществляется путем анализа собственных троек так, чтобы каждая группа соответствовала отдельной идентифицируемой компоненте ряда. Выбор нескольких ведущих собственных троек соответствует оптимальной аппроксимации временного ряда в соответствии с известным свойством оптимальности SVD.

Шаг 4: Восстановление (диагональное усреднение)

Диагональное усреднение (1.6), применяемое к результирующей матрице \mathbf{X}_{I_k} , дает *восстановленный ряд* $\tilde{\mathbf{X}}^{(k)} = (\tilde{x}_1^{(k)}, \dots, \tilde{x}_N^{(k)}) = \mathcal{T}^{-1} \circ \Pi_{\mathcal{H}}(\mathbf{X}^{(k)})$. Таким образом, исходный ряд (x_1, \dots, x_N) разлагается в сумму m восстановленных рядов:

$$x_n = \sum_{k=1}^m \tilde{x}_n^{(k)}, \quad n = 1, \dots, N. \quad (1.14)$$

Восстановленный ряд, полученный в результате элементарной группировки, будем называть *элементарным восстановленным рядом*.

Если группировка разумна, то мы получим разумное разложение на идентифицируемые компоненты ряда. Типичными результирующими разложениями являются ‘сигнал плюс шум’ или ‘тренд плюс сезонность плюс шум’.

Так же как и в общей схеме, этапы 1 и 2 Basic SSA иногда объединяются в так называемый ‘этап разложения’, а этапы 3 и 4 — в ‘этап восстановления’.

1.8.2. Модель временных рядов

Временные ряды конечного ранга

Хотя метод SSA не требует задание модели временного ряда и поэтому SSA можно рассматривать как разведочный метод, существует модель, которая идеально подходит для SSA.

Мы говорим, что ряд имеет L -ранг r , если его L -траекторная матрица имеет ранг r . Ряд S называется рядом конечного ранга r ($\text{rank } S = r$), если $\text{rank}_L S = r$ для любой достаточно большой длины ряда N и длины окна L . Термин ‘бесконечный ранг’ применим для случая бесконечных рядов. При наличии исходного бесконечного временного ряда (обрезанного на некотором N) $L = L(N)$ может стремиться к бесконечности и в этом случае ранг траекторной матрицы, как правило, тоже стремится к бесконечности. Для временного ряда конечного ранга, ранг r траекторной матрицы конечен и фиксирован для достаточно больших N и L так, что $r \leq \min(L, N - L + 1)$.

Полезно знать ранг временного ряда и форму сингулярных векторов его траекторной матрицы, так как знание ранга означает знание числа компонентов SVD, которые мы должны сгруппировать для извлечения соответствующего компонента ряда, в то время как форма сингулярных векторов вместе со свойствами собственных значений помогает в нахождении этих компонентов SVD.

За подробностями и полным описанием мы обращаемся к [1, Глава 5]. Здесь мы упомянем, что экспоненциальный ряд $s_n = Ae^{\alpha n}$, $n = 1, 2, \dots$, имеет ранг 1, линейная функция $s_n = an + b$, $a \neq 0$, имеет ранг 2, синусоида с $s_n = A \sin(2\pi\omega n + \phi)$ имеет ранг 2 для $0 < \omega < 0.5$ и ранг 1 для $\omega = 0.5$. Сингулярные векторы траекторных матриц временного ряда имеют ту же форму, что и сам ряд, что следует из того факта, что строки и столбцы траекторных матриц являются подрядами исходного ряда. Эта информация помогает при выборе групп на этапе группировки.

Линейные рекуррентные соотношения, характеристические многочлены и корни

Временные ряды конечного ранга тесно связаны с рядами, управляемыми линейными рекуррентными соотношениями (ЛРС). В частности, для бесконечных временных рядов, класс временных рядов, управляемых ЛРС, совпадает с классом временных рядов конечного ранга.

Определение 1. Временной ряд $S_N = (s_i)_{i=1}^N$ является управляемым ЛРС, если существуют a_1, \dots, a_t такие, что

$$s_{i+t} = \sum_{k=1}^t a_k s_{i+t-k}, \quad 1 \leq i \leq N-t, \quad a_t \neq 0, \quad t < N-1. \quad (1.15)$$

Число t называется порядком ЛРС, а a_1, \dots, a_t – коэффициентами ЛРС. Если $t = r$ является

минимальным порядком ЛРС, которое управляет временным рядом S_N , то соответствующее ЛРС называется *минимальным*.

Минимальное ЛРС единственno и его порядок равен рангу ряда.

Как было сказано в разделе 1.5, хорошо известно, что временной ряд $S_\infty = (s_1, \dots, s_n, \dots)$ удовлетворяет ЛРС (1.15) для всех $i \geq 0$ тогда и только тогда, когда

$$s_n = \sum_{m=1}^p \left(\sum_{j=0}^{k_m-1} C_{mj} n^j \right) \mu_m^n, \quad (1.16)$$

где комплексные коэффициенты C_{mj} зависят от первых t точек s_1, \dots, s_t .

Для вещественных временных рядов (1.16) подразумевает, что класс временных рядов, управляемых ЛРС, состоит из сумм произведений многочленов, экспонент и синусоид

$$s_n = \sum_{m=1}^{\tilde{p}} \left(\sum_{j=0}^{k_m-1} \tilde{c}_{mj} n^j \right) e^{\alpha_m n} \cos(2\pi\omega_m n + \phi_m), \quad 0 \leq \omega_m \leq 0.5. \quad (1.17)$$

Минимальное ЛРС определяет все параметры кроме C_{mj} в (1.16) и все параметры кроме \tilde{c}_{mj} и ϕ_m в (1.17).

Определение 2. Многочлен $P_t(\mu) = \mu^t - \sum_{k=1}^t a_k \mu^{t-k}$ называется *характеристическим многочленом ЛРС* (1.15).

Корни характеристического многочлена называются *характеристическими корнями* соответствующего ЛРС. Корни характеристического многочлена минимального ЛРС, управляющего рядом, которые можно назвать *сигнальные корни ЛРС* или *характеристические корни ряда*, определяют значения параметров μ_m и k_m в (1.16) следующим образом. Пусть временной ряд $S_\infty = (s_1, \dots, s_n, \dots)$ удовлетворяет ЛРС (1.15) с $a_t \neq 0$ и $i \geq 1$. Рассмотрим характеристический многочлен ЛРС (1.15) и обозначим его различные (комплексные) корни через μ_1, \dots, μ_p , где $p \leq t$. Все эти корни ненулевые, так как $a_t \neq 0$, причем k_m — кратность корня μ_m ($1 \leq m \leq p$, $k_1 + \dots + k_p = t$).

Пусть $C_{k_p-1,j} \neq 0$ для всех j , это соответствует случаю минимального ЛРС. Тогда ранг временного ряда S_∞ , заданного в (1.16), равен $r = \sum_{m=1}^p k_m$. В вещественно-значном случае, если $\tilde{c}_{k_p-1,j} \neq 0$ для всех j , то ранг временного ряда S_∞ , заданного в (1.17) равен $r = \sum_{m=1}^{\tilde{p}} \delta_m k_m$, где $\delta_m = 1$ для ω_m равного 0 или 0.5 и $\delta_m = 2$ для $0 < \omega_m < 0.5$.

Если мы найдем сигнальные корни $\mu_m = \rho_m e^{\pm i 2\pi \omega_m}$ характеристического многочлена ЛРС, управляющего сигналом, то мы можем оценить параметры сигнала. Например, частота ω_m экспоненциально модулированной синусоиды может быть найдена с помощью аргумента соответствующих сопряженных корней, в то время как модуль корня ρ_m дает экспоненциальный показатель $\alpha_m = \ln \rho_m$.

1.8.3. Разделимость и выбор параметров

Понимание разделимости очень важно для понимания того, как работает SSA. Напомним, что если два временных ряда $X_N^{(1)}$ и $X_N^{(2)}$ являются разделимыми, то $X_N^{(1)}$ может быть извлечен из наблюдаемого ряда $X_N = X_N^{(1)} + X_N^{(2)}$. Это означает, что существует разбиение на группы при шаге группировки такое, что $\tilde{X}_N^{(m)} = X_N^{(m)}$.

Давайте определим разделимость формально. Пусть $\mathbf{X}^{(m)}$ — траекторные матрицы компонентов рассматриваемого ряда, $\mathbf{X}^{(m)} = \sum_{i=1}^{d_m} \sqrt{\lambda_{m,i}} U_{m,i} V_{m,i}^T$, $m = 1, 2$, — их SVD. Напомним, что пространства столбцов и строк траекторных матриц называются столбцовыми и строковыми траекторными пространствами соответственно.

Определение 3. Пусть L фиксировано. Два ряда $X_N^{(1)}$ и $X_N^{(2)}$ называются слабо разделимыми (или просто разделимыми), если их столбцовые траекторные пространства ортогональны и то же самое справедливо для строковых траекторных пространств; то есть, $(\mathbf{X}^{(1)})^T \mathbf{X}^{(2)} = \mathbf{0}_{K,K}$ и $\mathbf{X}^{(1)} (\mathbf{X}^{(2)})^T = \mathbf{0}_{L,L}$.

Определение 4. Два ряда $X_N^{(1)}$ и $X_N^{(2)}$ называются сильно разделимыми, если они слабо разделимы и множества сингулярных значений их L -траекторных матриц не совпадают; то есть $\lambda_{1,i} \neq \lambda_{2,j}$ для любых i и j .

Слабая разделимость означает, что на шаге разложения существует такое SVD, которое позволяет сделать правильную группировку. Возможность существования не разделяющего сингулярного разложения, которое не допускает правильной группировки, связана с неединственностью SVD в случае равных сингулярных значений. Сильная разделимость означает, что любое сингулярное разложение траекторной матрицы допускает правильную группировку. Поэтому, чтобы быть уверенным, что SSA производит точное разделение, мы должны потребовать сильную (приближенную) разделимость.

По определению, слабая разделимость означает ортогональность пространств столбцов и строк траекторных матриц компонентов рядов $X_N^{(1)}$ и $X_N^{(2)}$. Для приближенной (асимптотической) разделимости с $\tilde{X}_N^{(m)} \approx X_N^{(m)}$, нам необходимо условие приближенной (асимптотической) ортогональности подрядов рассматриваемых компонент. Асимптотическая разделимость рассматривается в случае, если L и/или K стремятся к бесконечности.

Для достаточно длинных временных рядов SSA может приблизенно разделить, например, сигнал и шум, синусоиды с разными частотами, тренд и сезонность ([1, Глава 6, Раздел 1.5] и [29, Раздел 2.3.3]).

Продемонстрируем разделимость двух синусоид с разными частотами ω_1 и ω_2 : $x_n^{(i)} = A_i \cos(2\pi\omega_i n + \phi_i)$. Эти синусоиды асимптотически слабо разделимы; то есть, их подряды асимптотически ортогональны, если их длины стремятся к бесконечности. Однако скорость сходимости зависит от разницы между частотами. Если частоты близки, а длина временного ряда недостаточно велика, два ряда могут быть далеки от ортогональности и, следовательно,

не разделимы. Обратим внимание, что две синусоиды с одинаковыми амплитудами асимптотически слабо разделимы, но не сильно разделимы, и поэтому смешиваются в разложениях SSA.

Мера разделимости

Так называемая \mathbf{w} -корреляционная матрица содержит очень полезную информацию, которая может быть использована для обнаружения разделимости и идентификации групп. Эта матрица состоит из взвешенных косинусов углов между восстановленными компонентами временного ряда.

Пусть w_n ($n = 1, 2, \dots, N$) — это веса, определенные в (1.6): w_n равно количеству раз, когда элемент ряда x_n появляется в траекторной матрице. Определим \mathbf{w} -скалярное произведение временных рядов длины N как $(\mathbf{Y}_N, \mathbf{Z}_N)_{\mathbf{w}} = \sum_{n=1}^N w_n y_n z_n = \langle \mathbf{Y}, \mathbf{Z} \rangle_F$, где \mathbf{Y} и \mathbf{Z} — L -траекторные матрицы рядов \mathbf{Y}_N и \mathbf{Z}_N соответственно. Получаем частный случай (1.8) \mathbf{w} -корреляции между \mathbf{Y}_N и \mathbf{Z}_N для временных рядов:

$$\rho_{\mathbf{w}}(\mathbf{Y}_N, \mathbf{Z}_N) = (\mathbf{Y}_N, \mathbf{Z}_N)_{\mathbf{w}} / (\|\mathbf{Y}_N\|_{\mathbf{w}} \|\mathbf{Z}_N\|_{\mathbf{w}}).$$

Хорошо разделенные компоненты в (1.14) имеют слабую (или нулевую) корреляцию в то время как плохо разделенные компоненты обычно имеют высокую корреляцию. Поэтому, глядя на матрицу \mathbf{w} -корреляций между элементарными восстановленными рядами $\tilde{\mathbf{X}}_N^{(i)}$ и $\tilde{\mathbf{X}}_N^{(j)}$ можно найти группы коррелированных компонентов ряда и использовать эту информацию для последующей группировки. Одним из основных правил является: “не включать сильно коррелированные компоненты в разные группы”. \mathbf{w} -Корреляции также могут быть использованы для проверки сгруппированного разложения.

Очень информативно изобразить матрицу абсолютных значений \mathbf{w} -корреляций между компонентами ряда графически в черной-белой шкале в виде матрицы, где малые корреляции показаны белым цветом, а корреляции, абсолютные значения которых близки к 1, показаны черным цветом; см., например, рис. 1.6.

Выбор параметров

Условия (приближенной) разделимости дают рекомендации по выбору длины окна L : она должна быть достаточно большой ($L \sim N/2$) и если мы хотим извлечь периодическую компоненту с известным периодом, то длина окна, кратная периоду, обеспечивает лучшую разделимость [1, Раздел 1.6]. Выбор параметров обсуждается в разделе 3.4. Если мы выберем несколько ведущих собственных троек, то SSA с малым L выполняет сглаживание ряда как фильтр порядка $2L - 1$, см. [29, Раздел 3.9]. В целом, выбор длины окна имеет большое значение, но результат обычно стабилен по отношению к небольшим изменениям в значениях L .

Если временной ряд имеет сложную структуру, то рекомендуется так называемый последовательный SSA [29, Раздел 2.5.5]. Последовательный SSA состоит из двух этапов; на первом этапе, тренд извлекается с небольшой длиной окна, а затем периодические компоненты обнаруживаются и извлекаются из остатка с $L \sim N/2$.

Обоснование

Если мы используем SSA как разведочную технику, не требующую задания модели временного ряда, то обоснование разложения не может быть формальным; оно должно быть основано на теории разделимости и интерпретируемости результатов. Обработка методом SSA в режиме реального времени или в пакетном режиме возможна, если класс рядов не слишком широк и хорошо определен, чтобы можно было зафиксировать правило для выбора подходящих параметров. Для проведения статистического тестирования необходимо определить модель временного ряда.

1.8.4. Разложение экспоненциально-модулированных гармоник

Так как далее будем рассматривать колебательный компонент временного ряда как сумму экспоненциально-модулированных гармоник, опишем свойства таких гармоник.

Элементы вещественного экспоненциально-модулированного гармонического ряда S имеют вид

$$s_n = a e^{\alpha n} \cos(2\pi\omega n + \phi), \quad (1.18)$$

где $0 \leq \phi < 2\pi$, $a \neq 0$, $0 < \omega \leq 0.5$, $\sin \phi \neq 0$ для $\omega = 0.5$.

Как и прежде, L — длина окна, $K = N - L + 1$, $\Lambda^{(1)}$ и $\Lambda^{(2)}$ — подпространства строк и столбцов траекторной матрицы ряда S_N .

L -ранг d ряда — это ранг его траекторной матрицы и, следовательно, количество левых или правых сингулярных векторов, соответствующих ненулевым сингулярным числам траекторной матрицы \mathbf{X} .

Предложение 1. [1, Раздел 5.1]

1. L -ранг d ряда S , элементами которого являются (1.18) равен 1, если $\omega = 0.5$; в остальных случаях $d = 2$.
2. Если $\omega \neq 0.5$, то подпространства $\Lambda^{(1)}$ и $\Lambda^{(2)}$ имеют базисы

$$\begin{aligned} &\{(1, e \cos(2\pi\omega), \dots, e^{\alpha(L-1)} \cos(2\pi(L-1)\omega))^T, \\ &(0, e \sin(2\pi\omega), \dots, e^{\alpha(L-1)} \sin(2\pi(L-1)\omega))^T\} \quad \text{и} \\ &\{(1, e \cos(2\pi\omega), \dots, e^{\alpha(K-1)} \cos(2\pi(K-1)\omega))^T, \\ &(0, e \sin(2\pi\omega), \dots, e^{\alpha(K-1)} \sin(2\pi(K-1)\omega))^T\} \end{aligned}$$

соответственно.

3. В случае $\omega = 0.5$, одномерные $\Lambda^{(1)}$ и $\Lambda^{(2)}$ являются линейными линейными оболочками векторов $(1, c, \dots, c^{L-1})^T$ и $(1, c, \dots, c^{K-1})^T$ соответственно, где $c = -e^\alpha$.

Предложение 2. [33, Предложение 2.3] Пусть $\alpha = 0$, $\omega \neq 0.5$ и $L\omega$ и $K\omega$ – целые числа. Тогда собственные значения SVD для ряда S с элементами вида (1.18) одинаковы и имеют вид $\lambda_1 = \lambda_2 = a^2 LK/4$.

Из предложения 1 следует, что для $\omega \neq 0.5$ у ряда S существует в точности два левых сингулярных вектора U_1 и U_2 , соответствующих ненулевым собственным значениям траекторной матрицы, и что элементы векторов U_1 и U_2 могут быть представлены в следующем виде:

$$u_k^{(1)} = a_1 e^{\alpha k} \cos(2\pi\omega k + \phi_1), \quad u_k^{(2)} = a_2 e^{\alpha k} \cos(2\pi\omega k + \phi_2), \quad (1.19)$$

где $1 \leq k \leq L$, $0 \leq \phi_1, \phi_2 < 2\pi$, $a_1, a_2 \neq 0$.

Предложение 3. Пусть $\alpha = \alpha_N = C/N$, где C – некоторая константа, и $L = [\beta N]$, где $0 < \beta < 1$. Тогда для сингулярных векторов U_1 и U_2 ряда S , элементы которого имеют вид (1.19), выполняются соотношения $\lim_{L \rightarrow \infty} |\phi_1 - \phi_2| = \pi/2 (\text{ mod } \pi)$ и $\lim_{L \rightarrow \infty} (a_1/a_2) = 1$.

1.8.5. Алгоритм

В разделе 1.8.1 мы описали базовый метод SSA. Здесь мы формально представляем алгоритм Basic SSA. Отметим, что пакет RSSA эффективно реализует алгоритмы. (см. раздел 6.2 для краткого обсуждения). Поскольку эффективная реализация сложна и скрывает смысл шагов алгоритма, мы записываем алгоритмы в оригинальном виде.

Входными данными для всего алгоритма Basic SSA являются длина окна и способ группировки элементарных компонент \mathbf{X}_i в (1.12). Однако правило группировки используется после того, как произведено разложение (1.12). Поэтому группировка становится входными данными для этапа восстановления. По этой причине мы разделили алгоритм на две части. Обратим внимание, что модификации Basic SSA в основном отличаются только этапом разложения;

АЛГОРИТМ 1.1: Basic SSA: разложение

Input: Временной ряд \mathbf{X} длины N , длина окна L .

Output: Разложение траекторной матрицы в сумму элементарных матриц $\mathbf{X} = \mathbf{X}_1 + \dots + \mathbf{X}_d$, где $d = \text{rank } \mathbf{X}$ и $\mathbf{X}_i = \sqrt{\lambda_i} U_i V_i^T$ ($i = 1, \dots, d$).

- 1: Построить траекторную матрицу $\mathbf{X} = \mathcal{T}_{\text{SSA}}(\mathbf{X})$.
- 2: Вычислить SVD $\mathbf{X} = \mathbf{X}_1 + \dots + \mathbf{X}_d$, $\mathbf{X}_i = \sqrt{\lambda_i} U_i V_i^T$.

Алгоритмы восстановления практически одинаковы для различных версий SSA; на входе у них разложение траекторной матрицы в сумму матриц единичного ранга и разбиение компонент единичного ранга на группы. Таким образом, мы сформулируем общий алгоритм восстановления и сделаем комментарии относительно специфических особенностей модификаций в соответствующих разделах. Особенностью Basic SSA является то, что входным разложением является SVD, т.е. биортогональное разложение на компоненты ранга один, упорядоченные по вкладу $\sigma_i^2 = \lambda_i$ так, что $\sigma_1 \geq \dots \geq \sigma_d$.

АЛГОРИТМ 1.2: Восстановление

Input: Разложение $\mathbf{X} = \mathbf{X}_1 + \dots + \mathbf{X}_d$, где $\mathbf{X}_i = \sigma_i U_i V_i^T$ и $\|U_i\| = \|V_i\| = 1$; разбиение индексов: $\{1, \dots, d\} = \bigsqcup_{j=1}^m I_j$.

Output: Разложение временного ряда \mathbf{X} на идентифицируемые компоненты $\mathbf{X} = \mathbf{X}_1 + \dots + \mathbf{X}_m$.

- 1: Построить сгруппированное матричное разложение $\mathbf{X} = \mathbf{X}_{I_1} + \dots + \mathbf{X}_{I_m}$, где $\mathbf{X}_I = \sum_{i \in I} \mathbf{X}_i$.
- 2: Вычислить $\mathbf{X} = \mathbf{X}_1 + \dots + \mathbf{X}_m$, где $\mathbf{X}_i = \mathcal{T}_{\text{SSA}}^{-1} \circ \Pi_{\mathcal{H}}(\mathbf{X}_{I_i})$.

1.8.6. Модификация Toeplitz SSA для стационарных временных рядов

Метод

Toeplitz SSA отличается от Basic SSA только шагом 2 общей схемы, представленной на рис. 1.3; то есть, в разложении \mathbf{X} на матрицы ранга один. Basic SSA использует сингулярное разложение на этом шаге с U_i , вычисленными как собственные векторы $\mathbf{S} = \mathbf{X}\mathbf{X}^T$. Если длина N ряда \mathbf{X} невелика и ряд предполагается стационарным, то обычная рекомендация состоит в том, чтобы заменить матрицу \mathbf{S} некоторой другой матрицей, которая построена в предположении стационарности.

Сначала отметим, что в Basic SSA мы можем рассматривать *лаг-ковариационную матрицу* $\mathbf{C} = \mathbf{S}/K$ вместо \mathbf{S} для получения сингулярного распределения траекторной матрицы \mathbf{X} . Действительно, собственные векторы матриц \mathbf{S} и \mathbf{C} одинаковы и собственные значения отличаются только на коэффициент $1/K$.

Обозначим через $c_{ij} = c_{ij}(N)$ элементы лаг-ковариационной матрицы \mathbf{C} . Если временной ряд стационарен с нулевым средним, L фиксировано и $K \rightarrow \infty$, то $\lim_{N \rightarrow \infty} c_{ij}(N) = R_{\mathbf{X}}(|i - j|)$ при $|i - j| \leq L$, где $R_{\mathbf{X}}(k)$ обозначает член автоковариационной временного ряда с лагом k . Поэтому мы можем определить теплицеву версию лаг-ковариационной матрицы, положив равные значения \tilde{c}_{ij} на каждой анти-диагонали матрицы $|i - j| = k$. Наиболее естественный способ определения значения \tilde{c}_{ij} и соответствующей матрицы $\tilde{\mathbf{C}}$ заключается в том, чтобы вычислить

$$\tilde{c}_{ij} = \frac{1}{N - |i - j|} \sum_{m=1}^{N - |i - j|} x_m x_{m+|i - j|}, \quad 1 \leq i, j \leq L. \quad (1.20)$$

При использовании этой формулы обычно предполагается, что временной ряд \mathbf{X} центрирован, т.е., среднее $\bar{x} = \sum_{i=1}^N x_i / N$ вычитается из всех $x_i \in \mathbf{X}$.

Пусть $L \leq K$ и обозначим через P_i ($i = 1, \dots, L$) собственные векторы $\tilde{\mathbf{C}}$; эти векторы образуют ортонормированный базис \mathbb{R}^L . Тогда разложение по элементарным матрицам может быть записано как $\mathbf{X} = \sum_{i=1}^L P_i (\mathbf{X}^T P_i)^T$. Упорядочивание слагаемых осуществляется по величине $\sigma_i = \|\mathbf{X}^T P_i\|$. Заметим, что такое упорядочение в общем случае отличается от упорядочения собственных значений матрицы $\tilde{\mathbf{C}}$, соответствующих собственным векторам P_i . Некоторые из этих собственных значений могут быть даже отрицательными, так как матрица $\tilde{\mathbf{C}}$ не обязательно положительно определенная.

Если исходный ряд стационарен с нулевым средним, то использование *лаг-ковариационной теплицевой матрицы* $\tilde{\mathbf{C}}$ может быть более подходящим, чем использование лаговой ковариационной матрицы \mathbf{C} . С другой стороны, Toeplitz SSA не подходит для нестационарных рядов; если исходный ряд имеет существенную нестационарную компоненту, то Basic SSA работает лучше, чем Toeplitz SSA. Например, если мы имеем дело с чисто-экспоненциальным рядом, то он описывается единственной собственной тройкой для любой длины окна, в то время как Toeplitz SSA производит L собственных векторов для длины окна L ; более того, собственные векторы в Toeplitz SSA имеют некоторыми особыми свойствами [106]. Аналогичный эффект имеет место для линейных рядов, экспоненциально-модулированных синусоид и т.д.

Заметим, что использование \mathbf{C} для получения сингулярного разложения траекторной матрицы иногда называют ‘BK’, следуя [60], в то время как использование $\tilde{\mathbf{C}}$ для получения собственных векторов инициируется спектральным анализом и называется ‘VG’ вслед за [63]

Алгоритм

Алгоритм 1.3: Toeplitz SSA: разложение

Input: Временной ряд \mathbf{X} длины N , длина окна L .

Output: Разложение траекторной матрицы на элементарные матрицы $\mathbf{X} = \mathbf{X}_1 + \dots + \mathbf{X}_L$, где

$$\mathbf{X}_i = \sigma_i P_i Q_i^T, \quad \|P_i\| = \|Q_i\| = 1.$$

1: Строим траекторную матрицу $\mathbf{X} = \mathcal{T}_{\text{SSA}}(\mathbf{X})$.

2: Получаем разложение

$$\mathbf{X} = \sum_{i=1}^L \sigma_i P_i Q_i^T = \mathbf{X}_1 + \dots + \mathbf{X}_L, \quad (1.21)$$

где $\{P_i\}_{i=1}^L$ — собственные векторы матрицы $\tilde{\mathbf{C}}$, элементы которой вычисляются по (1.20), $S_i = \mathbf{X}^T P_i$, $Q_i = S_i / \|S_i\|$ и $\sigma_i = \|\mathbf{X}_i\|_F = \|S_i\|$. Компоненты упорядочены по величине σ_i : $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_L$.

Алгоритм восстановления точно такой же, как Алгоритм 1.2 с векторами (P_i, Q_i) , замененными на (U_i, V_i) .

1.8.7. Пример идентификации и разложения

Давайте продемонстрируем на простом примере, как визуально определить компоненты SSA и получить разложение SSA. На одномерных графиках (рис. 1.4) можно найти медленно меняющуюся компоненту (ET 1), в то время как на двумерных диаграммах (рис. 1.5) регулярные многоугольники говорят о парах синусоидальных компонент (ET 2–3 для $T = 12$, ET 4–5 для $T = 4$, ET 6–7 для $T = 6$, ET 8–9 для $T = 2.4$ и ET 10–11 для $T = 3$). На этих рисунках $U_i(k)$ обозначает k -ю координату собственного вектора i , полученного на этапе сингулярного разложения в SSA. Номера собственных векторов указаны в заголовках графиков. Рис. 1.6 с изображенными w -корреляциями дает предположение о группировке, поскольку сильно коррелированные компоненты должны быть включены в одну группу (черный цвет показывает корреляции, близкие к 1; белый цвет соответствует нулевой корреляции).

Полученный результат разложения на тренд, сезонность и шум показан на рис. 1.7.

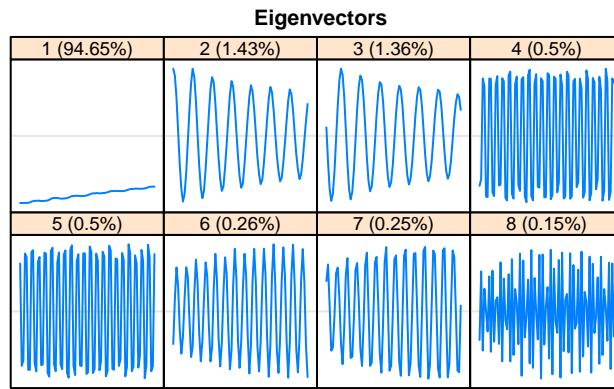


Рис. 1.4: “Fortified wines”, $L = 84$: одномерные графики собственных векторов $(k, U_i(k))$, $k = 1, \dots, L$.

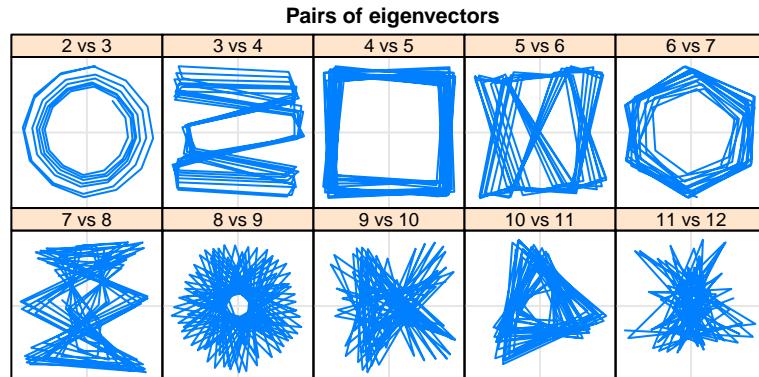


Рис. 1.5: “Fortified wines”, $L = 84$: двумерные диаграммы собственных векторов $(U_i(k), U_{i+1}(k))$, $k = 1, \dots, L$.

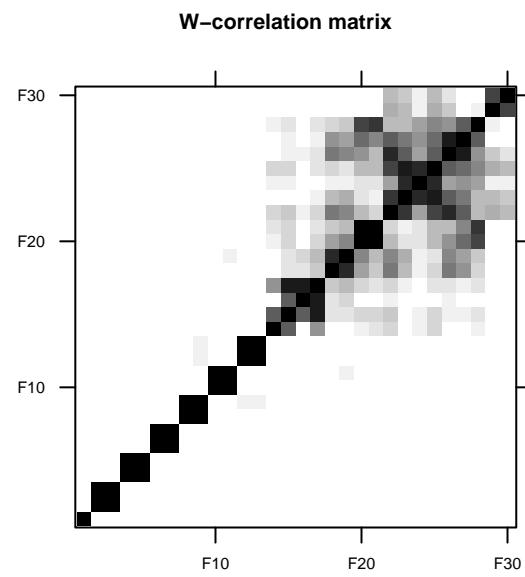


Рис. 1.6: “Fortified wines”, $L = 84$: w -корреляции между элементарными RC.

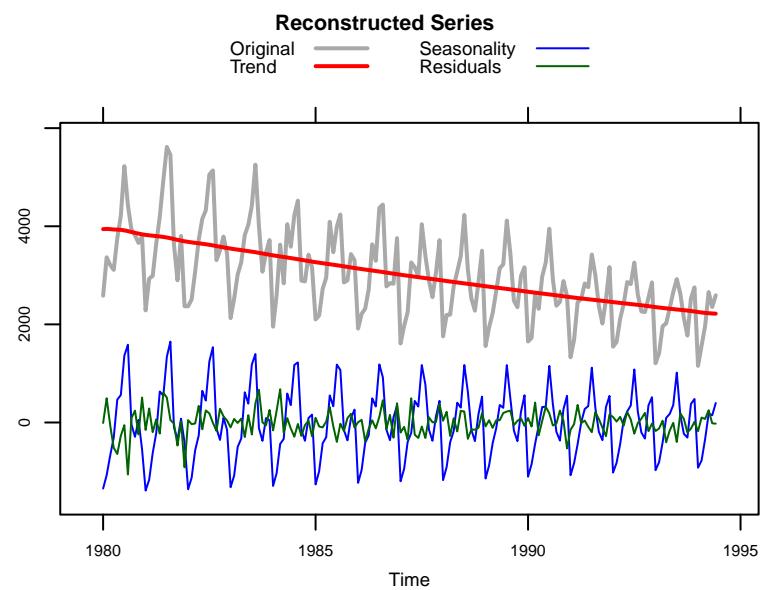


Рис. 1.7: “Fortified wines”, $L = 84$: разложение для групп ET1, ET2–11 и ET12–84.

1.8.8. Пример проблем с разделимостью

В разделе 1.8.7 мы рассмотрели пример с простым трендом и периодическими компонентами. Этот пример соответствует хорошей разделимости между трендом, сезонными и шумовыми компонентами. Однако для трендов сложной формы, которые часто встречаются в анализе временных рядов, существует большая вероятность того, что трендовые компоненты разложения SSA могут быть смешаны с сезонными компонентами. Ниже мы рассмотрим короткий временной ряд и продемонстрируем проблему отсутствия разделимости.

Проанализируем первые пять лет временного ряда ‘MotorVehicle’ (рис. 1.1). Первое разложение выполнено с помощью Basic SSA с $L = 24$. На рис. 1.8 изображены 11 элементарных восстановленных рядов (reconstructed components, RCs), т.е. временных рядов, которые получаются при группировке $\{1, \dots, d\} = \bigcup_{i=1}^d \{i\}$. Эти временные ряды полезны для рассмотрения при процедуре группировки, так как восстановление по собственным тройкам из группы I является просто суммой элементарных восстановленных компонент с номерами из I .

Видно, что тренд содержится в медленно меняющейся RC 1 и частично в RC 2, 8 и 9; последние элементарные восстановленные временные ряды содержат смесь тренда и сезонности. Это означает, что восстановление тренда с помощью ET1 недостаточно, в то время как восстановление, скажем, с помощью ET1,2 будет содержать сезонность (см. рис. 1.9).

Таким образом, если нет возможности отделить тренд от остатка, то невозможно извлечь точный тренд. Этот пример объясняет, почему очень важно пытаться добиться разделимости различными способами, такими как выбор длины окна, последовательный SSA, вложенные модификации Basic SSA (например, итеративный Oblique SSA, SSA с производными; см. раздел 2.2.1).

Давайте применим SSA с производными (DerivSSA), подробно рассмотренный в разделе 2.3, к группе из 11 ведущих ET. На рис. 1.10 четко видно, что трендовыми компонентами являются RC 9–11. На рис. 1.11 показан извлеченный точный тренд.

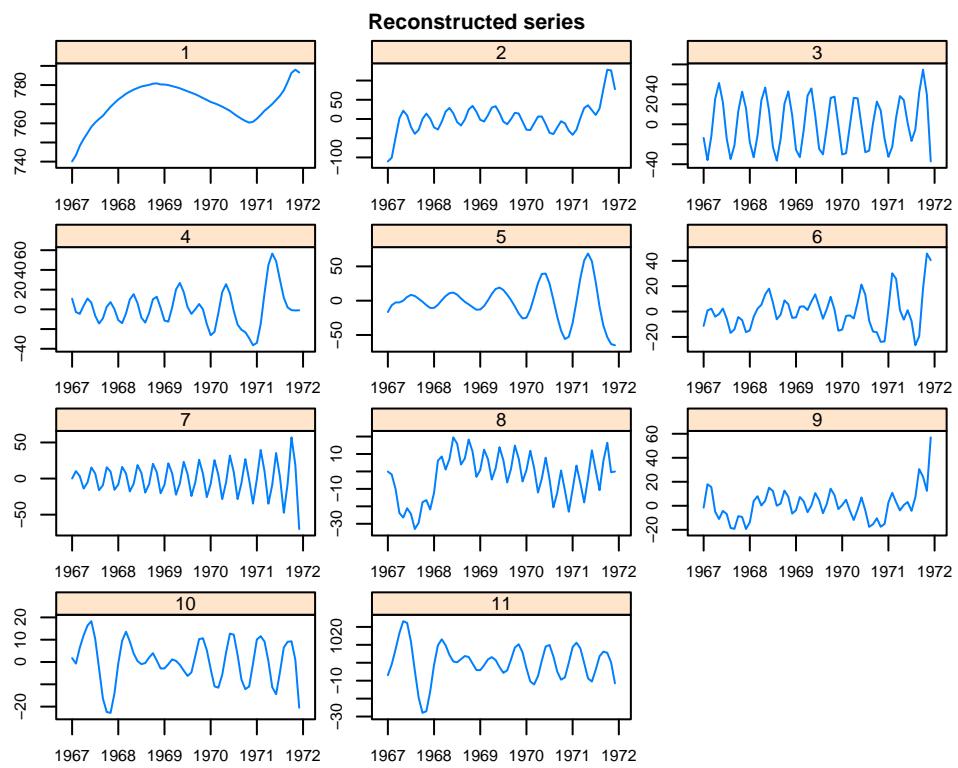


Рис. 1.8: ‘MotorVehicle’ (5 лет), SSA с $L = 24$: элементарные RC; плохая разделимость.

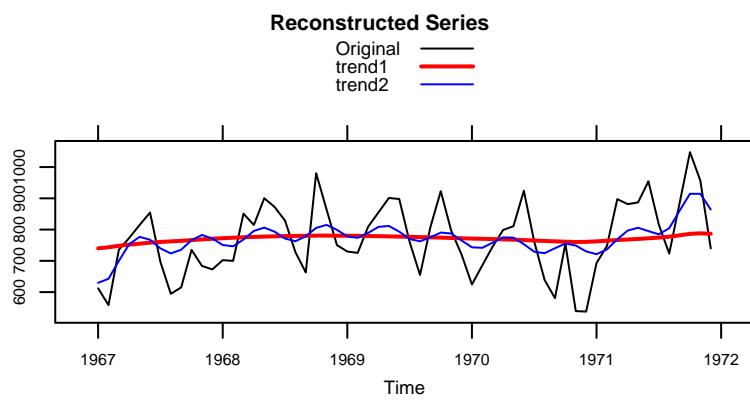


Рис. 1.9: ‘MotorVehicle’ (5 лет), SSA с $L = 24$: два восстановления тренда, ET1 и ET1–2; плохая разделимость.

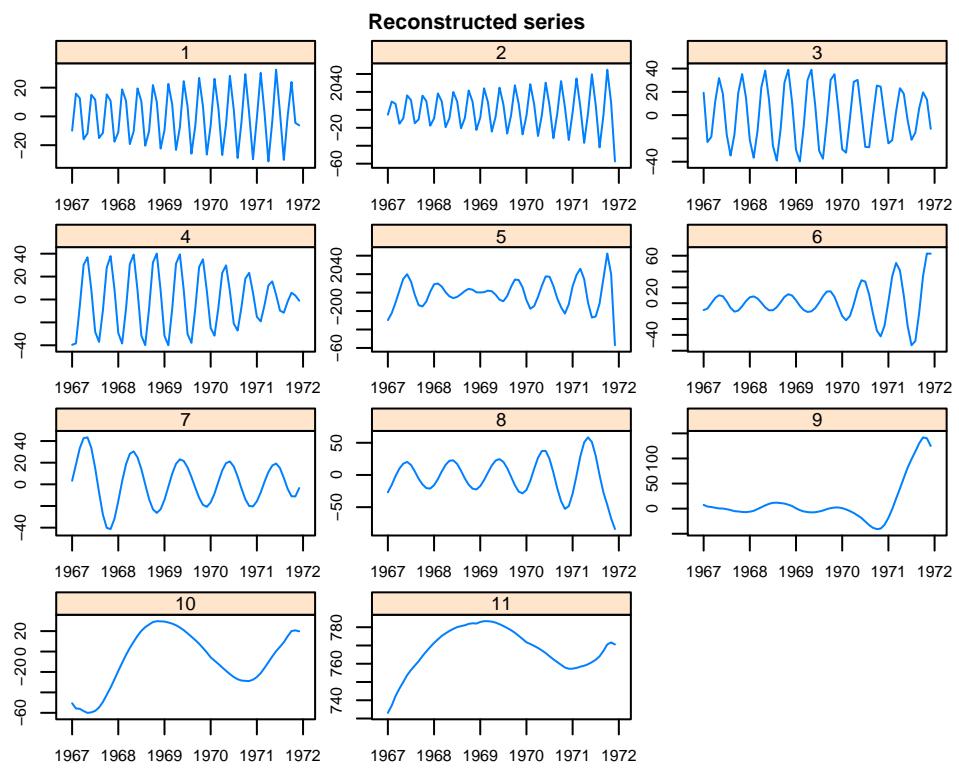


Рис. 1.10: ‘MotorVehicle’ (5 лет), DerivSSA с $L = 24$: элементарные RC; хорошая разделимость.

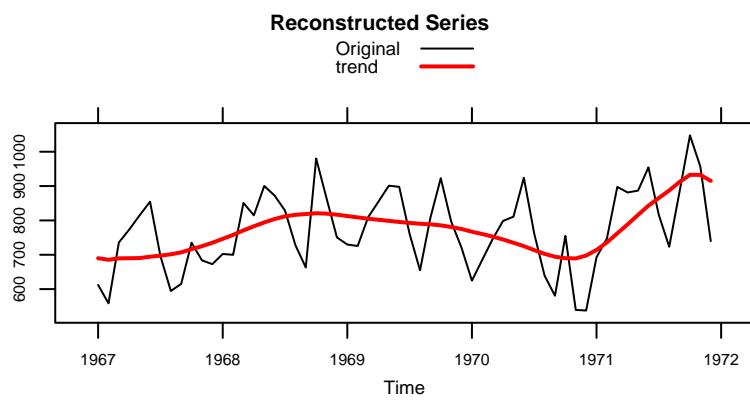


Рис. 1.11: ‘MotorVehicle’ (5 лет), DerivSSA с $L = 24$: восстановление тренда, ET9-11; хорошая отделимость.

1.9. Прогнозирование и оценка параметров для одномерных временных рядов

Во введении и разделе 1.8.2 описан класс сигналов, который является подходящим для метода SSA, и три формы его задания. Это сигналы, траекторная матрица которых имеет неполный ранг, или сигналы, управляемые ЛРС, или сигналы в явной параметрической форме (1).

Для таких сигналов корректно поставить задачу прогнозирования и оценки параметров. В данном разделе мы кратко опишем решение этих задач в рамках SSA.

1.9.1. Выделение сигнала через проекции

Рассмотрим частный случай $\mathbf{X} = \mathbf{S} + \mathbf{N}$, $\mathbf{X} = (x_1, \dots, x_N)$, где \mathbf{S} — сигнал ранга r , \mathbf{N} — шум, и зададим параметры: длину окна L и ранг сигнала r .

Введем две проекции по норме Фробениуса: $\Pi_r : \mathbb{R}^{L \times K} \mapsto \mathcal{M}_r$, где \mathcal{M}_r множество матриц ранга не больше r , и $\Pi_{\mathcal{H}} : \mathbb{R}^{L \times K} \mapsto \mathcal{H}$, где \mathcal{H} — множество ганкелевых матриц. Пусть \mathcal{T} определено в (1.1).

Схема SSA для извлечения сигнала:

$$\mathbf{X} \xrightarrow{\boxed{L}} \mathbf{X} = \begin{pmatrix} x_1 & x_2 & \cdots & x_K \\ x_2 & x_3 & \cdots & x_{K+1} \\ \vdots & \vdots & \ddots & \vdots \\ x_L & x_{L+1} & \cdots & x_N \end{pmatrix} \xrightarrow{\text{SVD: } (\sqrt{\lambda_m}, U_m, V_m), \quad \Pi_r} \boxed{r}$$

$$\begin{cases} \mathcal{L}_r = \text{span}(U_1, \dots, U_r) \\ \text{это пространство сигнала;} \\ \Pi_r \text{ это проектор на } \mathcal{L}_r; \\ \widehat{\mathbf{S}} = \sum_{m=1}^r U_m (\mathbf{X}^T U_m)^T = \Pi_r \mathbf{X}. \end{cases} \xrightarrow{\Pi_{\mathcal{H}}} \widetilde{\mathbf{S}} = \begin{pmatrix} \tilde{s}_1 & \tilde{s}_2 & \cdots & \tilde{s}_K \\ \tilde{s}_2 & \tilde{s}_3 & \cdots & \tilde{s}_{K+1} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{s}_L & \tilde{s}_{L+1} & \cdots & \tilde{s}_N \end{pmatrix} \xrightarrow{\mathcal{T}^{-1}} \widetilde{\mathbf{S}}.$$

Таким образом, краткая форма алгоритма SSA для выделения сигнала выглядит следующим образом

$$\widetilde{\mathbf{S}} = \mathcal{T}^{-1} \Pi_{\mathcal{H}} \Pi_r \mathcal{T} \mathbf{X}.$$

Полученная таким образом оценка, вообще говоря, не имеет ранг r . Для получения в пределе оценки сигнала ранга r используется метод Кэдзоу, который можно записать через попеременные проекции как

$$\widetilde{\mathbf{S}}_k = \mathcal{T}^{-1} (\Pi_{\mathcal{H}} \Pi_r)^k \mathcal{T} \mathbf{X}. \quad (1.22)$$

1.9.2. Оценка параметров

В этом разделе мы опишем, как оценить коэффициенты и параметры компонент ряда, управляемого ЛРС $s_n = \sum a_i s_{n-i}$. Мы будем предполагать, что метод SSA способен приближенно извлечь исследуемый компонент ряда, т.е., интересующая нас компонента приближенно выделена и длина окна вместе с модификацией SSA выбраны правильным образом.

Хотя ряд, управляемый ЛРС, может быть выражен в общей параметрической форме (1.10), часто рассматривается частный случай ряда в виде $S = (s_n)$ с $s_n = \sum_{i=1}^r C_i \mu_i^n$, $\mu_i \in \mathbb{C}$, или, в вещественно-значной форме, $s_n = \sum_{i=1}^p A_i \exp(\alpha_i n) \cos(2\pi n \omega_i + \phi_i)$, где A_i , α_i , ω_i и ϕ_i ($i = 1, \dots, p$) — неизвестные параметры, значения которых могут быть (и часто бывают) интересны исследователю. Таким образом, возникает задача оценки параметров.

Метод

В этом разделе будем описывать методы оценки параметров, которые называются методами, основанными на подпространстве, когда оценивается только подпространство компонентов ряда, а шаг восстановления алгоритма SSA не имеет значения.

Пусть набор индексов I соответствует интересующей нас компоненте в построенном разложении траекторной матрицы $\mathbf{X} = \sum_{i=1}^d \sigma_i P_i Q_i^\top$. Для простоты обозначений мы полагаем $I = \{1, 2, \dots, r\}$. Тогда оцениваемое подпространство будет $\tilde{\mathcal{S}} = \text{span}(P_1, \dots, P_r)$. Мы всегда считаем множество $\{P_i\}$, порождающее $\tilde{\mathcal{S}}$, ортонормированным, так как если нет, то мы можем ортонормировать его. Поскольку исходные векторы P_i могут быть линейно зависимыми (например, в методе SSA с проекцией), процедура ортогонализации может уменьшить количество векторов. Будем считать, что r равно количеству векторов после ортогонализации.

Мы рассматриваем два вида параметризации; во-первых, в виде управляющей ЛРС и, во-вторых, в форме (1.10). Соответственно, мы описываем, как оценить коэффициенты управляющей ЛРС и параметры (1.10).

Оценка управляющей ЛРС Траекторное пространство \mathcal{S} сигнала S , управляемого некоторой ЛРС, соответствует многим ЛРС. Более точно, любой вектор из \mathcal{S}^\perp с последней координатой -1 порождает такую ЛРС; другими словами, любой такой вектор из \mathcal{S}^\perp дает набор коэффициентов для линейной комбинации первых $L - 1$ координат вектора из \mathcal{S} для получения последней координаты; см. [1, Раздел 5] и [29, Глава 3] для подробного объяснения.

Среди всех этих ЛРС (генерирующих одно и то же траекторное пространство \mathcal{S}) существует наилучшее ЛРС с минимальной суммой квадратов коэффициентов (так называемое *ЛРС с минимальной нормой*). ЛРС с минимальной нормой подавляет возможные возмущения в начальных данных больше всех других ЛРС, что важно, если мы используем это ЛРС для генерации или продолжения ряда, на основе SSA-аппроксимации исходных данных.

Для выбранной длины окна L сигнальное подпространство $\mathcal{S} \subset \mathbb{R}^L$ и поэтому ЛРС с минимальной нормой имеет порядок $L - 1$. Для каждого вектора-столбца P_i из \mathbf{P}_r , обозначим

π_i последнюю координату P_i , $\underline{P}_i \in \mathbb{R}^{L-1}$ вектор P_i с удаленной последней координатой, и $v^2 = \sum_{i=1}^r \pi_i^2$. Тогда элементы вектора

$$\mathcal{R} = (a_{L-1}, \dots, a_1) = \frac{1}{1-v^2} \sum_{i=1}^r \pi_i \underline{P}_i \quad (1.23)$$

дают коэффициенты управляющего ЛРС с минимальной нормой: $s_n = \sum_{i=1}^{L-1} a_i s_{n-i}$.

Для оцениваемого подпространства $\tilde{\mathcal{S}}$, оцениваемое ЛРС вычисляется аналогичным образом, на основе ортонормированного базиса $\tilde{\mathcal{S}}$.

Оценка частот Пусть $X_N = S_N + R_N$, где $s_n = \sum_{j=1}^r C_j \mu_j^n$ и ряды S_N и R_N приближенно разделимы для заданной длины окна L . Как правило, корни сигнального характеристического многочлена управляющего ЛРС позволяют оценить параметры сигнала μ_j , $j = 1, \dots, r$ (см. раздел 1.8.2). Однако, ЛРС с минимальной нормой не является минимальной и поэтому мы должны как-то отличить сигнальные корни от посторонних корней. Обычно сигнальные корни ЛРС с минимальной нормой имеют максимальные модули (например, см. [107]). Поэтому можно найти корни ЛРС с минимальной нормой, расположить их в порядке убывания и взять первые r корней.

Однако порядок никогда не гарантирован. Поэтому методы, которые способны отделить сигнальные и посторонние корни, могут оказаться очень полезными.

Опишем один из таких методов под названием ESPRIT [108]. Этот метод существует в двух версиях, LS-ESPRIT и TLS-ESPRIT, где LS означает наименьшие квадраты, TLS — полные наименьшие квадраты (см., например, описание метода в [29, Раздел 3.8.2]). Другие названия: HSVD [109] и HTLS [110]. Здесь мы описываем версию LS (HSVD).

Определим $\{P_1, \dots, P_r\}$ ортонормированный базис оцениваемого подпространства интересующего нас компонента. Зададим $\mathbf{P}_r = [P_1 : \dots : P_r]$, и пусть $\underline{\mathbf{P}}_r$ — матрица с удаленной последней строкой, а $\overline{\mathbf{P}}_r$ — матрица с удаленной первой строкой. Тогда μ_i можно оценить по собственным значениям матрицы $\underline{\mathbf{P}}_r^\dagger \overline{\mathbf{P}}_r$, где \dagger , обозначает псевдообратную. Соответственно, оцененные частоты являются аргументами μ_i .

Заметим, что матрица \mathbf{P}_r обычно состоит из выбранных собственных векторов U_i в базовом алгоритме SSA. Однако, подходит любой базис подпространства, которое оценивает подпространство сигнала.

Упомянем простой и быстрый метод оценки частоты, который используется для идентификации собственных троек на этапе группировки. Два вектора $U^{(1)}$ и $U^{(2)}$, образующие ортогональный базис траекторного пространства экспоненциально-модулированного синуса, имеют одинаковые периоды, а их фазы отличаются примерно на $\pi/2$. Пусть A и B определяются как $a_n = \rho^n \sin(2\pi\omega n + \phi)$ и $b_n = \rho^n \cos(2\pi\omega n + \phi)$. Обозначим угол между векторами через \angle . Тогда $\omega = \angle\left(\begin{pmatrix} a_1 \\ b_1 \end{pmatrix}, \begin{pmatrix} a_2 \\ b_2 \end{pmatrix}\right)/(2\pi)$. Следовательно, мы можем оценить частоту, используя базисные векторы $U^{(1)}$ и $U^{(2)}$. Поскольку эти векторы не имеют точно такой же

формы, как A и B , можно рассмотреть последовательность углов $\angle \left(\begin{pmatrix} u_i^{(1)} \\ u_i^{(2)} \end{pmatrix}, \begin{pmatrix} u_{i+1}^{(1)} \\ u_{i+1}^{(2)} \end{pmatrix} \right) / (2\pi)$, $i = 1, \dots, L-1$, а затем взять среднее или медиану как оценку частоты; подробности см. в [1, Раздел 1.6]. В RSSA рассматривается медиана и медиана абсолютных отклонений от медианы, использующаяся как мера точности.

Алгоритмы

Хотя ЛРС, аппроксимирующее временной ряд, обычно используется для прогнозирования, оно также может быть полезно для построения модели сигнала. Поэтому мы представляем алгоритм для вычисления коэффициентов ЛРС с минимальной нормой.

АЛГОРИТМ 1.4: Оценка сигнальной ЛРС

Input: Матрица $\mathbf{P}_r \in \mathbb{R}^{L \times r}$, состоящая из ортонормированных векторов-столбцов, которые образуют базис подпространства оцениваемого сигнала.

Output: Коэффициенты $\mathcal{R} = (a_{L-1}, \dots, a_1)$ соответствующего ЛРС.

- 1: Для каждого вектора-столбца P_i из \mathbf{P}_r , вычислить π_i и \underline{P}_i , $v^2 = \sum_{i=1}^r \pi_i^2$. Если v^2 равно 1, то СТОП с сообщением об ошибке “Коэффициент вертикальности равен 1”.
- 2: Вычислить $\mathcal{R} = \frac{1}{1-v^2} \sum_{i=1}^r \pi_i \underline{P}_i$.

Следующий алгоритм показывает, как параметры μ_i в $s_n = \sum_{i=1}^r C_i \mu_i^n$ могут быть оценены по корням характеристического многочлена ЛРС, управляющего этим временным рядом (см. раздел 1.8.2 для описания связи между ЛРС и их характеристическими многочленами). Данное ЛРС является оценкой ЛРС, управляющего рядом ранга r ; поэтому только корни r соответствуют сигналу, а остальные корни являются посторонними. Поскольку часто (но не всегда!) сигнальные корни ЛРС с минимальной нормой имеют большие модули, чем посторонние корни, мы можем выбрать сигнальные корни с большими абсолютными значениями среди всего набора корней.

АЛГОРИТМ 1.5: Оценка сигнальных корней через характеристический многочлен ЛРС

Input: Коэффициенты $A = (a_1, \dots, a_m)$ ЛРС сигнала $s_n = \sum_{i=1}^m a_i s_{n-i}$, ранг r .

Output: Сигнальные корни μ_i , $i = 1, \dots, r$.

- 1: Построить характеристический многочлен $P(\mu) = \mu^d - \sum_{i=1}^m a_i \mu^{n-i}$.
- 2: Вычислить корни μ_1, \dots, μ_m многочлена $P(\mu)$.
- 3: Упорядочить корни так, чтобы $|\mu_1| \geq \dots \geq |\mu_m|$.
- 4: Ведущие корни μ_i , $i = 1, \dots, r$, являются кандидатами в сигнальные корни.

Алгоритм 1.6 является одним из наиболее известных алгоритмов оценки высокого разрешения среди алгоритмов оценки частот и коэффициентов затухания на основе подпространства.

АЛГОРИТМ 1.6: ESPRIT

Input: Матрица $\mathbf{P}_r \in \mathbb{R}^{L \times r}$, состоящая из ортонормированных векторов-столбцов, которые формируют базис пространства оцениваемого сигнала.

Output: r корней в форме (ρ_i, ω_i) .

- 1: Используя метод LS или TLS, найти матрицу $\mathbf{M} \in \mathbb{R}^{r \times r}$, удовлетворяющую $\overline{\mathbf{P}}_r \approx \underline{\mathbf{P}}_r \mathbf{M}$. Для LS-метода, $\mathbf{M} = \underline{\mathbf{P}}_r^\dagger \overline{\mathbf{P}}_r$.
 - 2: Найти собственные значения μ_i , $i = 1, \dots, r$, матрицы \mathbf{M} .
 - 3: Положить $\rho_i = \text{Mod}(\mu_i)$, $\omega_i = \text{Arg}(\mu_i)$.
-

Следующий алгоритм является дополнением к шагу Разложения, который используется для помощи при группировке синусоид с похожими частотами.

АЛГОРИТМ 1.7: Быстрая ('парная') оценка частот

Input: Два ортонормированных вектора $U^{(1)}$ и $U^{(2)}$, образующих оценку траекторного пространства синусоиды.

Output: Частота ω , период T .

- 1: Вычислить $\phi_i = \angle \left(\begin{pmatrix} u_i^{(1)} \\ u_i^{(2)} \end{pmatrix}, \begin{pmatrix} u_{i+1}^{(1)} \\ u_{i+1}^{(2)} \end{pmatrix} \right)$, $i = 1, \dots, L - 1$.
 - 2: Вычислить $\bar{\phi}$ как среднее или медиану $\{\phi_i\}$.
 - 3: $\omega = \bar{\phi}/(2\pi)$, $T = 1/\omega$.
-

1.9.3. Прогнозирование

Проблема прогнозирования — это проблема продолжения сигнала S , извлеченного из наблюдаемого ряда $X = S + R$. Для построения прогноза достаточно оценить траекторное пространство S , а затем построить прогнозируемый ряд на основе оцененного подпространства.

Очевидным способом прогнозирования является оценка параметров ряда и использование их для прогнозирования. Однако класс рядов, пригодных для прогнозирования, значительно шире, чем класс рядов для оценки параметров, и поэтому рассматриваемые ниже методы прогнозирования не оценивают параметры ряда, а лишь используют некоторые характеристики оцениваемого подпространства.

Как правило, прогнозируемый ряд должен иметь структуру для прогнозирования. В

рамках SSA мы говорим, что S имеет структуру, если S управляется ЛРС. Однако очень важно отметить, что SSA-прогнозы имеют смысл для гораздо более широкого класса рядов, когда ЛРС дает адекватное описание структуры ряда только локально, а не глобально; последнее является требованием для оценки параметров. Например, как правило, тренд не удовлетворяет ЛРС на всем временном диапазоне, но может быть локально аппроксимирован гладким рядом, управляемым ЛРС. В частности, относительно надежные прогнозы могут быть сделаны для ряда, который может быть аппроксимирован рядом вида $s_n = \sum_{i=1}^r C_i(n) \mu_i^n$, $\mu_i \in \mathbb{C}$, где $C_i(n)$ — медленно меняющиеся функции от n .

В любой версии алгоритма прогнозирования предполагается, что ряд S приближенно разделен с R с помощью выбранной модификации метода SSA.

Метод

Методы прогнозирования SSA тесно связаны с построением ЛРС, описанных в разделе 1.8.2. Ниже мы описываем два алгоритма прогнозирования. Один из них непосредственно использует построенное ЛРС для прогнозирования, в то время как другой алгоритм делает это неявно. Оба алгоритма дают одинаковые прогнозы для рядов, управляемых ЛРС, если параметры SSA выбраны правильно.

Подход Мы будем использовать те же обозначения, что и в разделе 1.8.2.

Пусть $\{P_i\}$ — базис подпространства $\tilde{\mathcal{S}}$ из \mathbb{R}^L . Тогда мы можем поставить задачу прогнозирования в этом подпространстве.

Различные модификации SSA, описанные в главе 2 (за исключением SSA с проекцией) дают оценку базиса сигнального подпространства.

Если применяется некоторая модификация SSA и набор собственных троек $\{(\sigma_i, P_i, Q_i), i \in I\}$ выбран для восстановления, тогда $\tilde{\mathcal{S}} = \text{span}\{P_i, i \in I\}$. Множество векторов $\{P_i, i \in I\}$ не обязательно является ортонормированным базисом. Поэтому первым обязательным шагом является орто-нормализация набора векторов $\{P_i, i \in I\}$. Сделав этот шаг, мы можем построить алгоритмы прогнозирования, рассматривая ортонормированный базис в качестве входных данных. Для простоты мы будем нумеровать базисные векторы от 1 до r .

Подпространство $\tilde{\mathcal{S}}$ порождает коэффициенты линейной комбинации для восстановления последних координат векторов из $\tilde{\mathcal{S}}$ по их первым $L - 1$ координатам. Линейная комбинация, используемая для прогнозирования, имеет минимальную евклидову норму коэффициентов среди всех линейных комбинаций, соответствующих подпространству $\tilde{\mathcal{S}}$. Если $\tilde{\mathcal{S}}$ является в точности траекторным подпространством ряда, управляемого ЛРС, то линейная комбинация с минимальной нормой соответствует ЛРС с минимальной нормой. Более того, продолжение ряда в этом подпространстве единственно.

Преимущество минимальной нормы для прогнозирования ЛРС показано в следующем предложении.

Предложение 4. Пусть $X_N = S_N + P_N$, где P_N — стационарный белый шум с нулевым средним и дисперсией σ^2 , X и S — некоторые L -запаздывающие векторы рядов X_N и S_N соответственно и $C \in \mathbb{R}^{L-1}$. Тогда для $x = C^T \bar{S}$ и $\tilde{x} = C^T \bar{X}$ имеем $E\tilde{x} = x$ и $D\tilde{x} = \|C\|^2 \sigma^2$.

Доказательство непосредственно следует из равенства $D\sum_{i=1}^{L-1} c_i(y_i + \varepsilon_i) = D\sum_{i=1}^{L-1} c_i \varepsilon_i = \|C\|^2 \sigma^2$, где $C = (c_1, \dots, c_{L-1})^T$ и ε_i , $i = 1, \dots, L-1$, являются независимыми одинаково распределенными случайными величинами с нулевым средним и дисперсией σ^2 .

Если $X = X_K$ — последний запаздывающий вектор S_N , то $\tilde{x} = C^T \bar{X}_K$ можно рассматривать как формулу прогнозирования, примененную к зашумленному сигналу и $\|C\|^2$ регулирует дисперсию этого прогноза.

Следующее свойство ЛРС с минимальной нормой, которое было получено в [111], чрезвычайно важно для прогнозирования: все посторонние корни ЛРС с минимальной нормой лежат внутри единичного круга комплексной плоскости. Это свойство дает нам надежду, что в случае реальных рядов (когда есть возмущение и у коэффициентов ЛРС с минимальной нормой, и у соответствующих начальные данные) члены, связанные с посторонними корнями в (1.10), лишь незначительно влияют на прогноз. Более того, учитывая результаты, касающиеся распределения посторонних корней (см. [112, 107]), можно ожидать, что посторонние слагаемые погасят друг друга.

Однако, если подпространство ряда оценено приближенно, можно предложить несколько вариантов прогнозирования. Если оценка подпространства была достаточно точной, то различные версии прогнозирования версии будут близки. В противном случае они могут существенно отличаться.

Теперь мы формально опишем алгоритмы прогнозирования. Для подробного объяснения см. [1, Глава 2].

Рекуррентное прогнозирование Рекуррентное прогнозирование SSA осуществляется с помощью ЛРС с минимальной нормой, определенной в (1.23).

Рекуррентный метод прогнозирования может быть сформулирован следующим образом.

1. Временной ряд $Y_{N+M} = (y_1, \dots, y_{N+M})$ определяется следующим образом

$$y_i = \begin{cases} \tilde{x}_i & \text{для } i = 1, \dots, N, \\ \sum_{j=1}^{L-1} a_j y_{i-j} & \text{для } i = N+1, \dots, N+M. \end{cases} \quad (1.24)$$

2. Числа y_{N+1}, \dots, y_{N+M} образуют M членов рекуррентного прогноза.

Таким образом, рекуррентное прогнозирование осуществляется путем прямого использования прогноза по ЛРС с коэффициентами, взятыми из $\mathcal{R} = (a_{L-1}, \dots, a_1)$.

Замечание 1. Определим линейный оператор $\mathcal{P}_{\text{Rec}} : \mathbb{R}^L \mapsto \mathbb{R}^L$ по формуле

$$\mathcal{P}_{\text{Rec}} Z = \begin{pmatrix} \bar{Z} \\ \mathcal{R}^T \bar{Z} \end{pmatrix}, \quad (1.25)$$

где \bar{Z} состоит из последних $L - 1$ координат Z . Зададим

$$Y_i = \begin{cases} \tilde{X}_i & \text{для } i = 1, \dots, K, \\ \mathcal{P}_{\text{Rec}} Y_{i-1} & \text{для } i = K + 1, \dots, K + M. \end{cases} \quad (1.26)$$

Легко видеть, что матрица $\mathbf{Y} = [Y_1 : \dots : Y_{K+M}]$ является траекторной матрицей ряда Y_{N+M} . Поэтому (1.26) можно рассматривать как векторную версию (1.24).

Замечание 2. В рекуррентном прогнозировании вместо восстановленного ряда может быть взят исходный ряд в качестве исходных данных для прогнозирования ЛРС. Это может быть целесообразно только в том случае, если для прогнозирования выбираются ведущие компоненты. Этот вариант может уменьшить погрешность, вызванную неточностью восстановления, но волатильность прогнозов может возрасти.

Если ЛРС не минимально, то только r корней соответствуют сигналу. Остальные корни являются посторонними и могут влиять на прогноз. Посторонние корни с модулями больше 1 могут привести к нестабильности.

Векторное прогнозирование Пусть $\mathcal{L}_r = \text{span}(P_i, i \in I)$ и \hat{X}_i является проекцией запаздывающего вектора X_i на \mathcal{L}_r . Рассмотрим матрицу

$$\Pi = \underline{\mathbf{P}} \underline{\mathbf{P}}^T + (1 - v^2) \mathcal{R} \mathcal{R}^T, \quad (1.27)$$

где $\underline{\mathbf{P}} = [\underline{P}_1 : \dots : \underline{P}_r]$ и \mathcal{R} определено в (1.23). Матрица Π определяет линейный оператор, который выполняет ортогональное проектирование $\mathbb{R}^{L-1} \mapsto \underline{\mathcal{L}}_r$, где $\underline{\mathcal{L}}_r = \text{span}(\underline{P}_i, i \in I)$. Наконец, мы определяем линейный оператор $\mathcal{P}_{\text{Vec}} : \mathbb{R}^L \mapsto \mathcal{L}_r$ по формуле

$$\mathcal{P}_{\text{Vec}} Z = \begin{pmatrix} \Pi \bar{Z} \\ \mathcal{R}^T \bar{Z} \end{pmatrix}. \quad (1.28)$$

Векторный метод прогнозирования может быть сформулирован следующим образом.

1. В приведенных выше обозначениях определим векторы

$$Y_i = \begin{cases} \hat{X}_i & \text{для } i = 1, \dots, K, \\ \mathcal{P}_{\text{Vec}} Y_{i-1} & \text{для } i = K + 1, \dots, K + M + L - 1. \end{cases} \quad (1.29)$$

2. Построив матрицу $\mathbf{Y} = [Y_1 : \dots : Y_{K+M+L-1}]$ и усреднив ее по антидиагоналям, мы получим ряд $y_1, \dots, y_{N+M+L-1}$.

3. Числа y_{N+1}, \dots, y_{N+M} образуют M членов векторного прогноза.

В рекуррентном прогнозировании мы выполняем диагональное усреднение для получения восстановленного ряда, а затем применяем ЛРС. В алгоритме векторного прогнозирования эти шаги применяются в обратном порядке. Векторный прогноз, как правило, немного более стабилен. Текущая быстрая реализация векторного прогнозирования делает векторное прогнозирование сравнимым с рекуррентным прогнозированием по вычислительным затратам, см. раздел 4.3.2.

Алгоритмы

Пусть некоторая модификация SSA применяется к временному ряду \mathbf{X} и пусть для восстановления выбрана группа собственных троек $\{(\sigma_i, P_i, Q_i), i \in I\}$. Предложенные алгоритмы прогнозирования сформулированы для прогнозирования в подпространстве $\mathcal{L}_r = \text{span}\{P_i, i \in I\} \subset \mathbb{R}^L$. Для простоты мы предполагаем, что $I = \{1, \dots, r\}$ и векторы $P_i, i \in I$, ортонормированы. Заметим, что результаты прогноза не зависят от выбора базиса в \mathcal{L}_r .

АЛГОРИТМ 1.8: Рекуррентное SSA прогнозирование

Input: Временной ряд \mathbf{X} длины N , длина окна L , ортонормированная система векторов $\{P_i\}_{i=1}^r$, горизонт прогнозирования M .

Output: Прогнозные значения $(\tilde{x}_{N+1}, \dots, \tilde{x}_{N+M})$.

- 1: Построить вектор $\mathcal{R} = (a_{L-1}, \dots, a_1)^T$ коэффициентов ЛРС с минимальной нормой по алгоритму 1.4, примененному к $\{P_i, i \in I\}$.
- 2: Построить восстановленную матрицу $\widehat{\mathbf{X}} = \mathbf{P} \mathbf{P}^T \mathbf{X}$, где $\mathbf{P} = [P_1 : \dots : P_r]$, и восстановленный ряд $\widetilde{\mathbf{X}} = (\widetilde{x}_1, \dots, \widetilde{x}_N)$ по $\widetilde{\mathbf{X}} = \mathcal{T}_{\text{SSA}}^{-1} \circ \Pi_{\mathcal{H}}(\widehat{\mathbf{X}})$.
- 3: Рассчитать прогнозные значения, применяя ЛРС с минимальной нормой:

$$\tilde{x}_n = \sum_{i=1}^{L-1} a_i \tilde{x}_{n-i}, \quad n = N+1, \dots, N+M$$

Алгоритм 1.8 записан в виде, когда в качестве базы для прогнозирования берется восстановленный ряд. Если за основу прогнозирования берется исходный ряд, то на шаге 3 вместо \tilde{x}_n нужно взять x_n для $n = N-L+2, \dots, N$.

Алгоритм 1.8 строит рекуррентное прогнозирование вперед. Рекуррентное прогнозирование назад получается путем применения прогнозирования вперед к ‘перевернутому’ ряду.

Следующий алгоритм реализует алгоритм векторного прогнозирования, где применение ЛРС с минимальной нормой и операция ганкелизации выполняются в обратном порядке.

 АЛГОРИТМ 1.9: Векторное SSA-прогнозирование

Input: Временной ряд \mathbf{X} длины N , длина окна L , ортонормированная система векторов $\{P_i\}_{i=1}^r$, горизонт прогнозирования M .

Output: Прогнозные значения $(\tilde{x}_{N+1}, \dots, \tilde{x}_{N+M})$.

- 1: Получить вектор $\mathcal{R} = (a_{L-1}, \dots, a_1)^T$ коэффициентов ЛРС с минимальной нормой по алгоритму 1.4, примененному к $\{P_i, i \in I\}$.
- 2: Вычислить матрицу Π проекции, заданную в (1.27).
- 3: Построить восстановленную матрицу $\widehat{\mathbf{X}} = \mathbf{P}\mathbf{P}^T\mathbf{X}$, где $\mathbf{P} = [P_1 \dots : P_r]$.
- 4: Расширить восстановленную матрицу $\widehat{\mathbf{X}} = [\widehat{X}_1 : \dots : \widehat{X}_K]$ векторами столбцов:

$$\widehat{X}_n = \mathcal{P}_{\text{Vec}} \widehat{X}_{n-1} \quad \text{for } n = K + 1, \dots, K + M + L - 1,$$

где \mathcal{P}_{Vec} дано в (1.28) и использует Π и \mathcal{R} . Обозначим расширенную матрицу $\widehat{\mathbf{X}}_{\text{ext}} \in \mathbb{R}^{L \times (K+M+L-1)}$.

- 5: Получить расширенный восстановленный ряд $\widetilde{\mathbf{X}}_{\text{ext}} = (\tilde{x}_1, \dots, \tilde{x}_{N+M+L-1})$ as $\widetilde{\mathbf{X}}_{\text{ext}} = \mathcal{T}_{\text{SSA}}^{-1} \circ \Pi_{\mathcal{H}}(\widehat{\mathbf{X}}_{\text{ext}})$.
 - 6: Вернуть прогнозные значения $(\tilde{x}_{N+1}, \dots, \tilde{x}_{N+M})$.
-

Дополнительные $L - 1$ векторов \widehat{X}_n на шаге 4 вычисляются, чтобы сделать значения прогноза независимыми от горизонта прогнозирования.

В алгоритме 1.9 восстановленный ряд берется за основу для прогнозирования. Для векторного прогнозирования не имеет смысла использовать исходный ряд в качестве базы для прогнозирования.

Заметим, что в таком прямом виде алгоритм 1.9 имеет гораздо большие вычислительные затраты, чем алгоритм 1.8. Однако, быстрая реализация, описанная в разделе 4.3.2 для многомерных временных рядов и реализованная в RSSA, делает векторное прогнозирование таким же быстрым, как и рекуррентное.

Глава 2

Шаг разложения для одномерных временных рядов

В данной главе и в главе 3 мы будем подробно рассматривать использование SSA для одномерных данных. Эта глава полностью посвящена SSA-анализу таких данных для построения разложений временного ряда. Рассмотрение SSA для заполнения пропусков, оценки сигнала и его обнаружения отложено до главы 3. Основное различие между материалами этих двух глав заключается в использовании моделей. В данной главе использование моделей минимально; напротив, методология главы 3 основана на моделях.

В терминологии главы 1, SSA для одномерных данных следует называть 1D-SSA. Однако, для краткости, в этой главе мы будем называть его просто SSA. Входными данными SSA для всех алгоритмов этой главы является набор $X_N = (x_1, \dots, x_N)$ из N вещественных чисел, который можно представить как временной ряд. Пусть L ($1 < L < N$) — длина окна и $K = N - L + 1$.

Общими шагами всех модификаций алгоритмов SSA, рассмотренных в этой главе, являются процедура вложения на шаге 1, группировка на шаге 3 и диагональное усреднение, которое делает восстановление на шаге 4 (см. раздел 1.8.1 и рис. 1.3).

Таким образом, модификации затрагивают только шаг 2 с разложением траекторной матрицы.

Раздел 2.1 содержит вариацию шага разложения, которая может быть использована, если известна дополнительная информация о структуре временного ряда. В качестве примера приводится извлечение полиномиальных трендов.

Разделы 2.2, 2.3 и 2.4 предлагают вложенные разложения для улучшения разделимости.

Раздел 2.5 описывает методы автоматизации шага группировки, который работает, если разделенные компоненты временного ряда (приближенно) разделимы. Поэтому методы улучшения разделимости должны применяться до группировки.

2.1. SSA с проекцией

Мы рассматриваем комбинацию сингулярного спектрального анализа, который не нуждается в модели ряда, заданной заранее, и параметрического подхода на основе подпространств, который включает в себя проекции на подпространства, заданные заранее. При изложении результатов этого раздела мы следуем [16].

Цель — обеспечить дополнительную теоретическую поддержку SSA с двойным центрированием (ProjSSA(1,1)), которая была известна ранее [1], а также расширить область применения полу параметрических модификаций Basic SSA. Приложения методов связаны с извлечением полиномиальных трендов.

2.1.1. SSA с центрированием

Существуют модификации SSA, называемые SSA с центрированием [1, Разделы 1.7 и 6.3]. Они служат для лучшего разделения постоянных (SSA с однократным центрированием) и линейных (SSA с двойным центрированием) трендов. Первоначально SSA с однократным центрированием был создан по аналогии с анализом главных компонент, где столбцы матрицы данных (которые можно интерпретировать как строки траекторной матрицы ряда) центрируются перед применением SVD. Опишем подход с центрированием.

Рассмотрим временной ряд \mathbf{X} длины N , длину окна L , $K = N - L + 1$, траекторную матрицу \mathbf{X} ряда \mathbf{X} .

Общая форма рассматриваемой модификации этапа разложения может быть выражена как

1. Вычисление специальной матрицы $\mathbf{C}^{(\text{center})} = \mathbf{C}(\mathbf{X})$ на основе априорной информации.
2. Вычисление $\mathbf{X}^* = \mathbf{X} - \mathbf{C}^{(\text{center})}$.
3. Построение SVD: $\mathbf{X}^* = \sum_{i=1}^{d^*} \sqrt{\lambda_i^*} U_i^* (V_i^*)^T$.

Таким образом, мы имеем разложение $\mathbf{X} = \mathbf{C}^{(\text{center})} + \sum_{i=1}^{d^*} \sqrt{\lambda_i^*} U_i^* (V_i^*)^T$.

Обозначим $E_M = (1, \dots, 1)^T \in \mathbb{R}^M$ вектор единиц. Центрирование рассматривается в следующих формах:

- *Однократное центрирование по строкам*, когда $\mathbf{C}_{\text{row}}^{(\text{center})}(\mathbf{X}) = (\mathbf{X}E_K/K)E_K^T$, соответствует усреднению по рядам, то есть каждый элемент ряда $\mathbf{C}_{\text{row}}^{(\text{center})}$ состоит из среднего значения соответствующей строки траекторной матрицы.
- *Однократное центрирование по столбцам*, когда $\mathbf{C}_{\text{col}}^{(\text{center})}(\mathbf{X}) = E_L(E_L^T\mathbf{X}/L)$, соответствует усреднению по столбцам.
- *Двойное центрирование*, когда $\mathbf{C}_{\text{both}}^{(\text{center})} = \mathbf{C}_{\text{row}}^{(\text{center})} + \mathbf{C}_{\text{col}}^{(\text{center})}(\mathbf{X} - \mathbf{C}_{\text{row}}^{(\text{center})}(\mathbf{X}))$, соответствует усреднению как по строкам, так и по столбцам.

Заметим, что центрирование можно рассматривать как проекцию строк и/или столбцов \mathbf{X} на $\text{span}(E_K)$ или $\text{span}(E_L)$ соответственно, так как $E_K E_K^T$ и $E_L E_L^T$ — это именно матрицы операторов проекции. Следовательно, центрирование в SSA можно рассматривать как предварительное проектирование траекторной матрицы на заданное подпространство; остаточная матрица \mathbf{X}^* впоследствии раскладывается с помощью сингулярного или любого другого разложения.

2.1.2. SSA с проекцией

Давайте обобщим подход, описанный в разделе 2.1.1, на проекции на произвольные пространства.

Обозначения. Пусть $\Pi_{\text{col}} : \mathbb{R}^L \rightarrow \mathcal{L}_{\text{col}}$ и $\Pi_{\text{row}} : \mathbb{R}^K \rightarrow \mathcal{L}_{\text{row}}$ являются ортогональными проекторами, где $\mathcal{L}_{\text{col}} \in \mathbb{R}^L$ называется пространством проекций столбцов и $\mathcal{L}_{\text{row}} \in \mathbb{R}^K$ называется пространством проекций строк. Для любого $\mathbf{Y} \in \mathcal{M}_{L,t}$, обозначим $\Pi_{\text{col}}(\mathbf{Y})$ матрицу состоящую из столбцов, которые получаются из проекций столбцов \mathbf{Y} , при этом для любого $\mathbf{Y} \in \mathcal{M}_{t,K}$ обозначим $\Pi_{\text{row}}(\mathbf{Y})$ матрицу, состоящую из строк, которые получаются в результате проекций строк \mathbf{Y} .

Обозначим базис пространства проекций столбцов ($P_i, i = 1, \dots, p$) и базис пространства проекций строк ($Q_i, i = 1, \dots, q$), $\mathbf{P} = [P_1 : \dots : P_p]$, $\mathbf{Q} = [Q_1 : \dots : Q_q]$. Без потери общности мы предполагаем, что $\{P_i, i = 1, \dots, p\}$ и $\{Q_i, i = 1, \dots, q\}$ являются ортонормированными базисами \mathcal{L}_{col} и \mathcal{L}_{row} (в противном случае, мы можем выполнить ортонормирование).

Схема SSA с центрированием, которая описана в разделе 2.1.1, распространяется на произвольные проекции с названием ‘SSA с проекцией’ (ProjSSA), а именно, $\mathbf{C} = \Pi_{\text{col}}(\mathbf{X})$ для проекции столбцов, $\mathbf{C} = \Pi_{\text{row}}(\mathbf{X})$ для проекции строк и $\mathbf{C} = \Pi_{\text{both}}(\mathbf{X})$ для двойной проекции, где

$$\begin{aligned}\Pi_{\text{both}}(\mathbf{X}) &= \Pi_{\text{row}}(\mathbf{X}) + \Pi_{\text{col}}(\mathbf{X} - \Pi_{\text{row}}(\mathbf{X})) \\ &= \Pi_{\text{col}}(\mathbf{X}) + \Pi_{\text{row}}(\mathbf{X} - \Pi_{\text{col}}(\mathbf{X})) \\ &= \Pi_{\text{row}}(\mathbf{X}) + \Pi_{\text{row}}(\mathbf{X}) - (\Pi_{\text{col}} \circ \Pi_{\text{row}})(\mathbf{X}).\end{aligned}\quad (2.1)$$

Если базис столбцов или строк отсутствует (т.е, соответствующая проекция не должна выполняться), то мы формально зададим соответствующий проектор как нулевой оператор и будем использовать обозначение $\mathbf{C} = \Pi_{\text{both}}(\mathbf{X})$ для всех случаев.

Разложение на элементарные матрицы. Общая форма разложения в SSA с проекцией имеет вид

$$\mathbf{X} = \mathbf{C} + \sum_{i=1}^{d^*} \sqrt{\lambda_i^*} U_i^* (V_i^*)^T, \quad (2.2)$$

где $\mathbf{C} = \Pi_{\text{both}}(\mathbf{X})$ и $\sum_{i=1}^{d^*} \sqrt{\lambda_i^*} U_i^* (V_i^*)^T$ — это SVD матрицы $\mathbf{X} - \mathbf{C}$. Продемонстрируем, что матрица \mathbf{C} может быть представлена в виде суммы элементарных матриц ранга 1. Тогда (2.2) будет аналогичен результату разложения (1.12), выполняемого Basic SSA, и, следовательно, этап восстановления также будет аналогичен этапу Basic SSA.

Заметим, что $\Pi_{\text{col}}(\mathbf{Y}) = \mathbf{P}\mathbf{P}^T \mathbf{Y} = \sum_{i=1}^p P_i (\mathbf{Y}^T P_i)^T$ и $\Pi_{\text{row}}(\mathbf{Y}) = \mathbf{Y}\mathbf{Q}\mathbf{Q}^T = \sum_{i=1}^q (\mathbf{Y} Q_i) Q_i^T$ являются разложениями в суммы элементарных матриц ранга один. Поэтому $\mathbf{C} = \Pi_{\text{both}}(\mathbf{X})$ также можно разложить в сумму элементарных матриц, так как Π_{both} можно выразить как последовательное применение операторов проекции Π_{row} и Π_{col} , см. (2.1). Для двойной проекции это разложение зависит от порядка проекций; для определенности мы будем применять сначала проектор строк.

Таким образом, матрицу \mathbf{C} можно рассматривать как сумму $p+q$ элементарных матриц

вида $\sigma_i^{(c)} P_i \tilde{Q}_i^T$, $i = 1, \dots, p$, и $\sigma_i^{(r)} \tilde{P}_i Q_i^T$, $i = 1, \dots, q$ (некоторые из них могут быть нулевыми), где тройки $(\sigma_i^{(c)}, P_i, \tilde{Q}_i)$ и $(\sigma_i^{(r)}, \tilde{P}_i, Q_i)$ имеют тот же смысл, что и собственные. Поэтому разложение (2.2) может быть преобразовано к разложению в сумму $d^* + p + q$ элементарных матриц ранга один, которые по конструкции ортогональны относительно нормы Фробениуса $\|\cdot\|$. Как следствие, вклад проекционного члена \mathbf{C} в разложение равен $\|\mathbf{C}\|^2 / \|\mathbf{X}\|^2$;

Следующая лемма описывает свойства разложения (2.2).

Лемма 1. *Разложение (2.2) удовлетворяет следующим свойствам:*

1. $d^* \leq \text{rank } \mathbf{X}$;
2. $d^* \geq \text{rank } \mathbf{X} - (p + q)$; равенство имеет место, если P_i , $i = 1, \dots, p$, принадлежат оболочке столбцов \mathbf{X} а Q_i , $i = 1, \dots, q$, принадлежат оболочке строк \mathbf{X} .

Доказательство. В силу определения ортогональной проекции, $\mathbf{X}^* = \mathbf{X} - \mathbf{C} = (\mathbf{I}_L - \mathbf{P}\mathbf{P}^T)\mathbf{X}(\mathbf{I}_K - \mathbf{Q}\mathbf{Q}^T)$, где \mathbf{I}_M обозначает $M \times M$ единичную матрицу. Следовательно, $d^* \leq \text{rank } \mathbf{X}$. С другой стороны, $\text{rank } \mathbf{X}$ не превосходит число элементарных матриц $d^* + p + q$; равенство достигается, если строки и столбцы проектируемых матриц принадлежат пространствам строк и столбцов \mathbf{X} соответственно. \square

Алгоритм

Давайте соберем вместе шаги SSA с проекцией в виде алгоритмов, разбив весь алгоритм на разложение и восстановление.

АЛГОРИТМ 2.1: SSA с проекцией: разложение

Input: Временной ряд \mathbf{X} длины N , длина окна L , ортонормированный базис пространства проекции столбцов \mathbf{X} , базис пространства проекций столбцов $(P_i, i = 1, \dots, p)$ и ортонормированный базис пространства проекций строк $(Q_i, i = 1, \dots, q)$. Либо p , либо q могут быть нулевыми.

Output: Разложение траекторной матрицы по элементарным матрицам $\mathbf{X} = \mathbf{X}_1 + \dots + \mathbf{X}_d$, где $\mathbf{X}_i = \sigma_i U_i V_i^T$ — нулевая или матрица единичного ранга.

- 1: Построить траекторную матрицу $\mathbf{X} = \mathcal{T}_{\text{SSA}}(\mathbf{X})$.
- 2: Вычесть проекцию строк: $\mathbf{X}' = \mathbf{X} - \mathbf{C}_{\text{row}}$, где

$$\mathbf{C}_{\text{row}} = \Pi_{\text{row}}(\mathbf{X}) = \sum_{i=1}^q \sigma_i^{(r)} \tilde{P}_i Q_i^T,$$

$\sigma_i^{(r)} = \|\mathbf{X} Q_i\|$, $\tilde{P}_i = \mathbf{X} Q_i / \sigma_i^{(r)}$ если $\sigma_i^{(r)} > 0$; в противном случае \tilde{P}_i — нулевой вектор.

3: Вычесть проекцию столбцов: $\mathbf{X}^* = \mathbf{X}' - \mathbf{C}_{\text{col}}$, где

$$\mathbf{C}_{\text{col}} = \Pi_{\text{col}}(\mathbf{X}') = \sum_{i=1}^p \sigma_i^{(c)} P_i \tilde{Q}_i^T,$$

$\sigma_i^{(c)} = \|\mathbf{X}'^T P_i\|$, $\tilde{Q}_i = \mathbf{X}'^T P_i / \sigma_i^{(c)}$ если $\sigma_i^{(c)} > 0$; в противном случае \tilde{Q}_i — нулевой вектор.

4: Построить SVD матрицы \mathbf{X}^* : $\mathbf{X}^* = \sum_{i=1}^{d^*} \mathbf{X}_i^*$, где $\mathbf{X}_i^* = \sqrt{\lambda_i^*} U_i^* (V_i^*)^T$.

5: В результате $\mathbf{X} = \sum_{i=1}^d \mathbf{X}_i$, где $d = p + q + d^*$, $\mathbf{X}_i = \sigma_i^{(r)} \tilde{P}_i \tilde{Q}_i^T$ для $i = 1, \dots, q$, $\mathbf{X}_{i+q} = \sigma_i^{(c)} P_i \tilde{Q}_i^T$ для $i = 1, \dots, p$, и $\mathbf{X}_{i+p+q} = \sqrt{\lambda_i^*} U_i^* (V_i^*)^T$ для $i = 1, \dots, d^*$.

Чтобы завершить алгоритм SSA с проекцией, опишем алгоритм этапа восстановления.

АЛГОРИТМ 2.2: SSA с проекцией: восстановление

Input: Разложение $\mathbf{X} = \mathbf{X}_1 + \dots + \mathbf{X}_d$ и группировка $\{1, \dots, d\} = \bigsqcup_{j=1}^m I_j$, которая не разбивает первые $p + q$ компоненты проекции, где q и p — число строк и столбцов компонентов проекции.

Output: Разложение временного ряда на идентифицируемые компоненты $\mathbf{X} = \mathbf{X}_1 + \dots + \mathbf{X}_m$.

- 1: Построить сгруппированное разложение матрицы $\mathbf{X} = \mathbf{X}_{I_1} + \dots + \mathbf{X}_{I_m}$, где $\mathbf{X}_{I_i} = \sum_{i \in I} \mathbf{X}_i$.
 - 2: Вычислить $\mathbf{X} = \mathbf{X}_1 + \dots + \mathbf{X}_m$, где $\mathbf{X}_i = \mathcal{T}^{-1} \mathcal{H}(\mathbf{X}_{I_i})$.
-

Единственное существенное отличие от восстановления в Basic SSA состоит в том, что множество матриц \mathbf{X}_i , $i = 1, \dots, p + q$, которое получается путем проекции, должно быть включено в одну группу. Тогда разложение полученного ряда не зависит от выбранных базисов $\{P_i, i = 1, \dots, p\}$ и $\{Q_i, i = 1, \dots, q\}$.

Подходящий класс временных рядов

Для SSA с проекцией естественным вопросом является то, какие ряды сохраняются при проекции; то есть, для каких типов рядов с траекторной матрицей \mathbf{X} мы имеем $\Pi_{\text{col}}(\mathbf{X}) = \mathbf{X}$ для проекции столбцов, $\Pi_{\text{row}}(\mathbf{X}) = \mathbf{X}$ для проекции строк и $\Pi_{\text{both}}(\mathbf{X}) = \mathbf{X}$ для двойной проекции.

Следующая лемма является прямым следствием определения проекции.

Лемма 2. Пусть \mathcal{L}_{row} содержит пространство строк матрицы \mathbf{X} и \mathcal{L}_{col} содержит пространство столбцов матрицы \mathbf{X} . Тогда $\Pi_{\text{row}}(\mathbf{X}) = \mathbf{X}$ и $\Pi_{\text{col}}(\mathbf{X}) = \mathbf{X}$.

Например, из леммы 2 следует, что для сохранения экспоненциального ряда с $s_n = C\mu^n$ по SSA с проекцией столбцов, проекция столбцов должна быть выполнена на пространство, которое содержит $\text{span}((\mu, \mu^2, \dots, \mu^L)^T)$, а для сохранения линейной функции с $s_n = an + b$

для любого b и ненулевого a , проекция столбца должна быть выполнена на пространство, которое содержит $\text{span}((1, 1, \dots, 1)^T, (1, 2, \dots, L)^T)$.

Выведем условие, достаточное для того, чтобы $\Pi_{\text{both}}(\mathbf{X}) = \mathbf{X}$ выполнялось для общего случая двойной проекции.

Лемма 3. Пусть столбцы матрицы $\mathbf{W} \in \mathcal{M}_{K,q}$ принадлежат \mathcal{L}_{row} и столбцы матрицы $\mathbf{S} \in \mathcal{M}_{L,p}$ принадлежат \mathcal{L}_{col} . Тогда $\Pi_{\text{both}}(\mathbf{X}) = \mathbf{X}$ для

$$\mathbf{X} = \tilde{\mathbf{S}}\mathbf{W}^T + \widetilde{\mathbf{S}}\mathbf{W}^T \quad (2.3)$$

для любых $\tilde{\mathbf{S}} \in \mathcal{M}_{L,q}$ и $\widetilde{\mathbf{W}} \in \mathcal{M}_{K,p}$.

Proof. По предположению, $\Pi_{\text{row}}(\mathbf{AW}^T) = \mathbf{AW}^T$ для любой матрицы $\mathbf{A} \in \mathcal{M}_{t,q}$, где $t \geq 1$, а $\Pi_{\text{col}}(\mathbf{SB}^T) = \mathbf{SB}^T$ для любой матрицы $\mathbf{B} \in \mathcal{M}_{t,p}$. Тогда

$$\Pi_{\text{both}}(\mathbf{X}) = \tilde{\mathbf{S}}\mathbf{W}^T + \Pi_{\text{row}}(\widetilde{\mathbf{S}}\mathbf{W}^T) + \widetilde{\mathbf{S}}\mathbf{W}^T + \Pi_{\text{col}}(\widetilde{\mathbf{S}}\mathbf{W}^T) - \Pi_{\text{col}}(\Pi_{\text{row}}(\widetilde{\mathbf{S}}\mathbf{W}^T + \widetilde{\mathbf{S}}\mathbf{W}^T)) = \mathbf{X},$$

поскольку $\Pi_{\text{col}} \circ \Pi_{\text{row}} \equiv \Pi_{\text{row}} \circ \Pi_{\text{col}}$ по ассоциативности умножения матриц. \square

Легко проверить, что траекторная матрица линейного ряда удовлетворяет условиям леммы 3 для случая двойного центрирования. Однако для общего случая более удобен подход, основанный на характеристических корнях. Начнем с технической леммы.

Лемма 4. Для любого многочлена P_d степени d и для любых l и k таких, что $l+k=d-1$, можно построить следующее разложение:

$$P_d(i+j) = P_{l,d}(i,j) + P_{d,k}(i,j),$$

где $P_{\tau,\varkappa}(i,j)$ обозначает многочлен от i и j би-степени (τ, \varkappa) .

Proof. Эта лемма доказывается соответствующей группировкой мономов $C_{u,v}i^u j^v$, $u+v \leq d$, из $P_d(i+j)$. \square

Напомним, что ряд, управляемый некоторым ЛРС, имеет вид (1.10), определяемый корнями характеристического многочлена, называемыми характеристическими корнями.

Теорема 1. Пусть ряды $\mathbf{Y}^{(m)}$, $m=1,2$, управляются минимальными ЛРС порядка r_m , $\mathbf{Y}^{(m)}$ — их траекторные матрицы. Пусть $\{\mu_j; j=1, \dots, s\}$ — множество, содержащее характеристические корни обоих рядов. Предположим, что $\mathbf{Y}^{(m)}$, $m=1,2$, имеют характеристические корни μ_j , $j=1, \dots, s$, с кратностью $d_j^{(m)} \geq 0$, $\sum_{j=1}^s d_j^{(m)} = r_m$. Пусть Π_{col} — проектор на пространство столбцов \mathcal{C} матрицы $\mathbf{Y}^{(1)}$, Π_{row} — проектор на пространство строк \mathcal{R} матрицы $\mathbf{Y}^{(2)}$, Π_{both} задан в (2.1). Тогда $\Pi_{\text{both}}(\mathbf{X}) = \mathbf{X}$, если и только если множество характеристических корней ряда X состоит из корней μ_j , $j=1, \dots, s$, кратности $d_j \leq d_j^{(1)} + d_j^{(2)}$.

Proof. В силу линейности проекторов и линейной зависимости Π_{both} от Π_{row} и Π_{col} , достаточно доказать теорему для случая одного корня μ . Пусть $\mathbf{Y}^{(1)}$ имеет характеристический корень μ кратности p , $\mathbf{Y}^{(2)}$ имеет характеристический корень μ кратности q .

Таким образом, мы должны доказать, что $\Pi_{\text{both}}(\mathbf{X}) = \mathbf{X}$ тогда и только тогда, когда ряд \mathbf{X} имеет вид $x_k = P_t(k)\mu^k$, где $t \leq p+q-1$. Достаточно рассмотреть $t = p+q-1$.

По лемме 4

$$P_{p+q-1}(i+j)\mu^{i+j} = P_{p-1,p+q-1}(i,j)\mu^i\mu^j + P_{p+q-1,q-1}(i,j)\mu^i\mu^j.$$

Это означает, что (2.3) имеет место для $\mathbf{W} \in \mathcal{M}_{K,q}$ и $\mathbf{S} \in \mathcal{M}_{L,p}$ таких, что пространство столбцов \mathbf{W} совпадает с \mathcal{R} , а пространство столбцов \mathbf{S} совпадает с \mathcal{C} .

Так как размерность пространства траекторных матриц, сохраняемых проектором Π_{both} , равна $r = r_1 + r_2$, мы нашли все такие матрицы. На этом доказательство завершено. \square

Следствие 1. Пусть \mathbf{Y} – ряд размерности r , \mathbf{Y} – его траекторная матрица, Π_{row} – проектор на строковое траекторное пространство, Π_{col} – проектор на столбцов траекторное пространство. Рассмотрим ряд \mathbf{X} с $x_n = (an+b)y_n$. Тогда $\Pi_{\text{both}}(\mathbf{X}) = \mathbf{X}$.

Замечание 3. Заметим, что умножение временного ряда на $an+b$, где $a \neq 0$, означает, что кратность характеристических корней ряда увеличивается на 1.

Заметим, что формально множества $\{P_i, i = 1, \dots, p\}$ и $\{Q_i, i = 1, \dots, q\}$ могут быть произвольными. Однако, если модель ряда частично известна, то в контексте SSA это означает, что компонент временного ряда удовлетворяет ЛРС и нам известны его характеристические корни (см. Раздел 1.9.2). Поэтому, чтобы извлечь, например, синусоиду с помощью проекций, мы должны знать ее период, а чтобы извлечь экспоненциальный тренд, мы должны знать его показатель. Такие условия часто являются слишком ограничительными. Явным исключением является извлечение полиномиальных трендов степени k , когда существует единственный характеристический корень, равный 1 кратности $k+1$ и для получения траекторного пространства полиномиального тренда необходимо предположить только его степень.

Следствие 2. Пусть Π_{row} – проектор на траекторное пространство ряда многочлена степени l , Π_{col} – проекция на траекторное пространство столбцов многочлена степени k . Тогда для любого многочлена $\mathbf{X} = P_{l+k+1}$ степени $l+k+1$ имеем $\Pi_{\text{both}}(\mathbf{X}) = \mathbf{X}$.

Замечание 4. Из леммы 2 сразу следует, что в условиях следствия 2, для любого многочлена $\mathbf{X} = P_l$ степени l имеем $\Pi_{\text{row}}(\mathbf{X}) = \mathbf{X}$ и для любого многочлена $\mathbf{X} = P_k$ степени k имеем $\Pi_{\text{col}}(\mathbf{X}) = \mathbf{X}$.

Разделимость

Мы ожидаем, что если компонент временного ряда управляет минимальным ЛРС и это ЛРС известно, то компонент ряда может быть отделен с помощью подходящей версии

SSA с проекцией лучше, чем это можно сделать с помощью Basic SSA.

Используя понятие разделимости, мы можем сформулировать это улучшение следующим образом. Пусть $\mathbf{X} = \mathbf{X}^{(1)} + \mathbf{X}^{(2)}$. Мы будем говорить, что компонент временного ряда $\mathbf{X}^{(1)}$ выделен с помощью SSA с проекцией, если $\mathbf{X}^{(1)} = \mathbf{C}$, где \mathbf{C} равно $\Pi_{\text{row}}(\mathbf{X})$, $\Pi_{\text{col}}(\mathbf{X})$ или $\Pi_{\text{both}}(\mathbf{X})$, в зависимости от типа проекции.

Пусть $\mathbf{X}^{(1)}$ — ряд конечного ранга, $\mathbf{X} = \mathbf{X}^{(1)} + \mathbf{X}^{(2)}$. Аналогично [1, Раздел 6.3], где рассматриваются условия разделимости по SSA с центрированием, можно получить следующие условия разделимости.

1. Basic SSA:

$\mathbf{X}^{(1)}$ и $\mathbf{X}^{(2)}$ разделимы, если (если и только если, по определению) их пространства строк и столбцов ортогональны.

2. SSA с проекцией строки на пространство строк $\mathbf{X}^{(1)}$:

$\mathbf{X}^{(1)}$ и $\mathbf{X}^{(2)}$ разделимы, если их пространства рядов ортогональны.

3. SSA с проекцией столбца на пространство столбцов $\mathbf{X}^{(1)}$:

$\mathbf{X}^{(1)}$ и $\mathbf{X}^{(2)}$ разделимы, если их пространства столбцов ортогональны.

4. SSA с двойной проекцией на пространство строк и столбцов \mathbf{Y} , где $\mathbf{X}^{(1)}$ выражается через \mathbf{Y} как $x_n^{(1)} = (an + b)y_n$, $a \neq 0$:

$\mathbf{X}^{(1)}$ и $\mathbf{X}^{(2)}$ разделимы по SSA с двойной проекцией, если \mathbf{Y} и $\mathbf{X}^{(2)}$ разделяются Basic SSA.

Заметим, что разделимость по SSA с проекцией всегда сильная, так как проекции на линейные пространства однозначно определены.

Для приближенной разделимости, где $\mathbf{X}^{(1)} \approx \mathbf{C}$, необходима приближенная ортогональность. Кроме того, асимптотическая разделимость может быть рассмотрена по аналогии с обычной разделимостью для Basic SSA и SSA с центрированием.

Напомним, что обычное двойное центрирование в SSA соответствует постоянному ряду \mathbf{Y} и, следовательно, линейному ряду $\mathbf{X}^{(1)}$. Ортогональность к постоянному ряду является гораздо более слабым условием, чем к линейному (более того, условие ортогональности к линейному ряду никогда не может быть выполнено в точности). В частности, любая синусоида с частотой ω асимптотически разделима с линейным трендом и имеет место точная разделимость по SSA с проекцией, если $L\omega$ и $K\omega$ — целые числа, то есть если L и K кратны периоду синусоиды. Поэтому для выделения линейных трендов рекомендуется двойное центрирование.

В случае полиномиального тренда степени больше 1 условия точной разделимости вообще не могут быть выполнены, даже для SSA с двойной проекцией. Однако мы все же можем ожидать, что в случае полиномиальных трендов, SSA с двойной проекцией также будет работать лучше, чем SSA только с проекцией строк или столбцов, а также лучше, чем Basic SSA. Это будет проверено в следующем разделе.

2.1.3. Примеры

Представленные примеры связаны с нахождением полиномиальных трендов. Для удобства, если проекции строк и столбцов выполняются на подпространства, порожденные многочленами степени $q - 1$ и $p - 1$ соответственно, то обозначим метод как ProjSSA(q,p). Напомним (см. следствие 2 и замечание 4), что выбор ProjSSA(q,p) соответствует извлечению полиномиального тренда степени $q + p - 1$. Нулевое значение для p или q означает, что соответствующая проекция не выполняется. Например, для выделения линейного тренда можно использовать ProjSSA(2,0) и ProjSSA(1,1). В ProjSSA(q,p), проекционная часть разложения, т.е. разложение матрицы $\mathbf{C} = \Pi_{\text{both}}(\mathbf{X})$, состоит из $p + q$ матриц ранга один.

Заметим, что реализация SSA с проекцией в RSSA является эффективной, поскольку использует подход, описанный в [113]. Вычислительные затраты на этапе разложения для общего случая составляют $\mathcal{O}((q+p)N \log N + qpN)$ для вычисления проекции с использованием быстрой свертки и $\mathcal{O}((q+p+r+\log N)Nr)$ для самого SVD разложения. Здесь r — количество необходимых ведущих собственных троек SVD, которое на практике невелико. Вычислительные затраты на этапе восстановления точно такие же, как и на этапе разложения Basic SSA.

SSA с проекцией и регрессией

Продемонстрируем, что обычная линейная регрессия и SSA с двойным центрированием, т.е. ProjSSA(1,1), используют разные постановки решаемой задачи и поэтому могут давать разные результаты. Это хорошо видно на коротких временных рядах. Для длинных временных рядов результаты очень близки. Также, в модели линейной регрессии с белым шумом, решение регрессии наименьших квадратов является наилучшей линейной несмещенной оценкой, см. теорему Гаусса-Маркова. Поэтому, чтобы продемонстрировать разницу, мы рассмотрим временной ряд, который содержит сезонный компонент.

Здесь мы рассматриваем временной ряд ‘Gasoline’, взятый из [114] и содержащий данные по спросу на бензин, ежемесячно, январь 1960 – июнь 1967, Онтарио, миллионы галлонов.

Рассмотрим первые два года и применим линейную регрессию и ProjSSA(1,1) с $L = 12$. Чтобы показать разницу, продолжим линию линейной регрессии с помощью оцененных коэффициентов. В RSSA реализован метод прогнозирования для SSA с проекцией. Однако, поскольку этот метод прогнозирования не был тщательно протестирован и доказан, мы его не используем. Построим прогноз с помощью линейной регрессии примененной к восстановлению, которое выполняется ProjSSA(1,1). Заметим, что процедура прогнозирования из RSSA дает аналогичный прогноз. В качестве эталона рассматривается линейная регрессия, построенная по всему ряду.

На рис. 2.1 видно, что линейный тренд ProjSSA(1,1) (черная тонкая линия) очень близок к линейному тренду, построенному по всему временному ряду (серая штриховая линия). Линия линейной регрессии (линия с круговыми точками) дает гораздо худшее приближение к тренду. Это объясняется следующими причинами. Подход через метод наименьших квад-

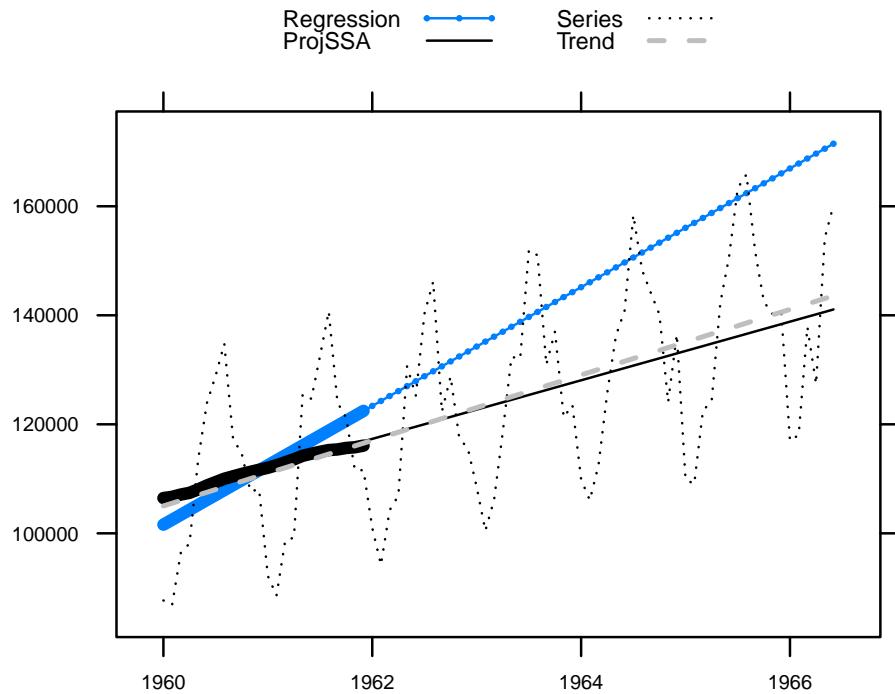


Рис. 2.1: ‘Gasoline’: SSA с проекцией, определение линейного тренда

ратов к оценке линейной регрессии минимизирует ошибку предсказания и поэтому сезонный компонент может сместить тренд линейной регрессии. Для ProjSSA(1,1) сезонная составляющая хорошо отделена от линейного тренда, так как для выбранных параметров $L = K = 12$ кратно сезонному периоду 12.

SSA с проекцией и Basic SSA

Пример, представленный в этом разделе, демонстрирует, что и SSA с проекцией, и Basic SSA могут извлекать тренды одинаковым образом. Рассмотрим пример ‘co2’ (Концентрация CO₂ в атмосфере Мауна Лоа, 468 наблюдений, ежемесячно с 1959 по 1997 гг. по 1997 год [115]).

Мы начинаем с извлечения линейного тренда и поэтому выбираем ProjSSA(1,1) для выполнения SSA с двойным центрированием.

По аналогии с SSA, большая длина окна помогает извлечь разделимые компоненты ряда, в то время как малая длина окна соответствует сглаживанию. Поэтому мы берем $L = 228$, кратное 12 и близкое к половине длины временного ряда, чтобы добиться лучшей разделимости, и небольшое значение $L = 36$ для сглаживания ряда. Три из четырех вариантов извлеченных трендов, представленных на рис. 2.2, почти совпадают.

Для выбора $L = 228$, извлеченный тренд близок к линейному, см. рис. 2.2 (слева вверху). Конечно, точный тренд ряда ‘co2’ не является линейным. Однако, компоненты проекции могут быть дополнены третьим и шестым компонентами SVD (ET5,8) для улучшения тренда

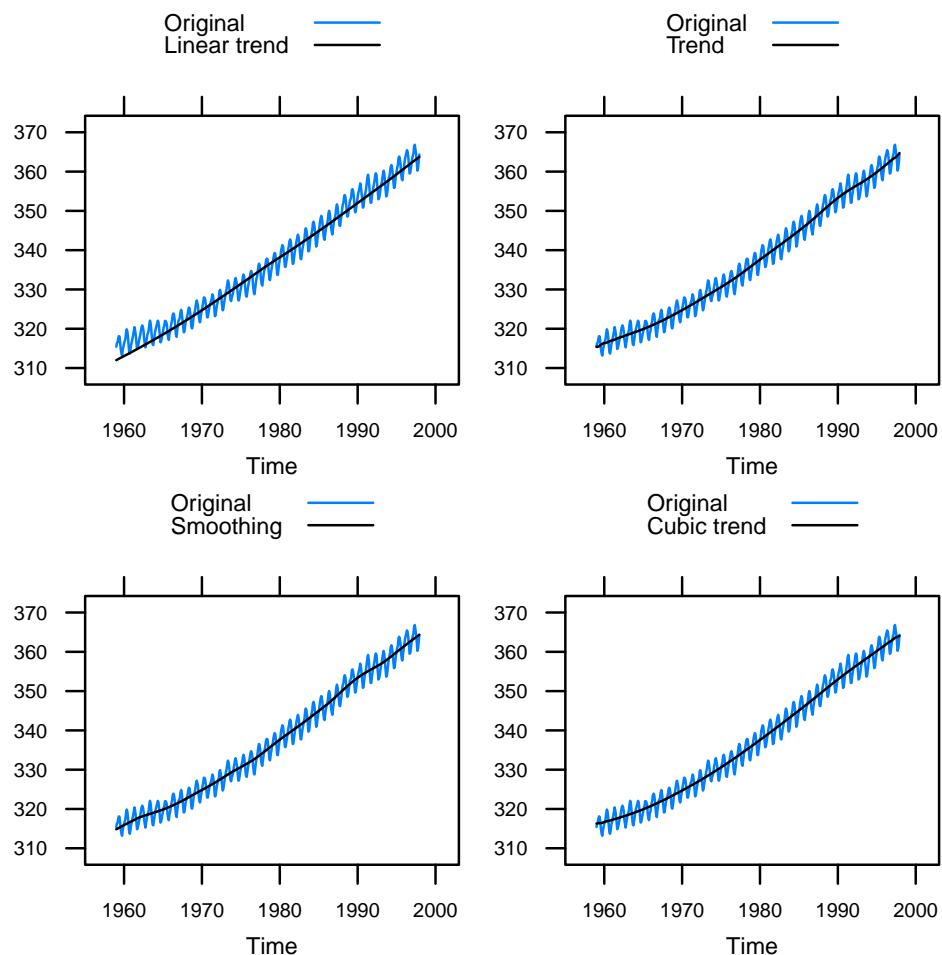


Рис. 2.2: ‘co2’: Восстановление тренда. Слева-сверху: ProjSSA(1,1), $L = 228$; справа-сверху: ProjSSA(1,1), $L = 228$, добавленные ET 5 и 8; слева-внизу: ProjSSA(1,1), $L = 36$; справа-внизу: ProjSSA(2,2), $L = 228$.

(рис. 2.2 (справа-сверху)). Рис. 2.2 (слева-внизу) показывает результат сглаживания с $L = 36$. Наконец, результат ProjSSA(2,2) с $L = 228$, который предназначен для извлечения кубического тренда, изображен на рис. 2.2 (справа-снизу). Извлеченный тренд очень похож на тренд в [8], который был извлечен с помощью Basic SSA (не показано).

Идентификация компонентов в разложении, полученном SSA с проекцией, точно такая же, как и в Basic SSA.

Численное сравнение

Реальные примеры, представленные в разделах 2.1.3 и 2.1.3, показывают, что результаты Basic SSA, SSA с проекцией и линейной регрессии могут быть как разными, так и похожими. Чтобы понять, какой метод лучше, проведем численное исследование.

Рассмотрим временной ряд длиной $N = 199$ с общим членом

$$x_n = t_n + s_n + \varepsilon_n, \quad (2.4)$$

где t_n — тренд, $s_n = A \sin(2\pi n \omega + \phi)$, ε_n — гауссовский белый шум со стандартным отклонением σ .

Для полученных оценок $\hat{t}_n^{(i)}$, где i — номер ряда с i -ой реализацией шума $\varepsilon_n^{(i)}$, $i = 1, \dots, M$, вычислим среднеквадратичную ошибку (RMSE) как $\sqrt{\frac{1}{MN} \sum_{i=1}^M \sum_{n=1}^N (\hat{t}_n^{(i)} - t_n)^2}$.

Линейный тренд и синусоида. Начнем со случая без шума ($\sigma = 0$) и поэтому возьмем $M = 1$. Пусть $t_n = an + b$. Зафиксируем $a = 1$, $b = -100$, $A = 1$ и будем менять ω с 0.02 до 0.1 (т.е., период изменяется с 50 до 10).

Поскольку результат метода наименьших квадратов сильно зависит от формы остатка, мы рассматриваем значения фазы $\phi = 0$ и $\phi = \pi/2$.

Рис. 2.3 (слева) содержит значения RMSE в случае $\phi = 0$ для Basic SSA с восстановлением по ET1–2, ProjSSA(2,0), ProjSSA(1,1) с $L = 100$, и для линейной регрессии. Видно, что худшие случаи для ProjSSA(1,1) примерно равны лучшим случаям для линейной регрессии.

В разделе 2.1.3 мы выполнили прогнозирование с помощью линейной регрессии, примененной к восстановленному тренду. Такая линейная регрессия, которая применяется к восстановленному тренду, может рассматриваться как другая оценка тренда. Рис. 2.3 (справа) содержит RMSE для линий линейной регрессии, построенных таким образом; в легенде добавлено ‘regr’. Обратим внимание, что толстая линия ‘линейной регрессии’ одинакова на левом и правом графиках, которые изображены в разных масштабах. Упорядочение методов SSA в целом одинаково, при этом методы SSA становятся лучше, чем линейная регрессия. Возможно, 0 — одно из худших значений ϕ для линейной регрессии.

Теперь рассмотрим $\phi = \pi/2$ как один из лучших случаев для линейной регрессии. Поведение ошибок совершенно иное (рис. 2.4 (слева)). Однако, точность ProjSSA(1,1) все еще лучше, чем у линейной регрессии. Линейная аппроксимация по методу наименьших квадра-

тов восстановленного по SSA тренда значительно улучшает точность методов SSA (рис. 2.4 (справа)).

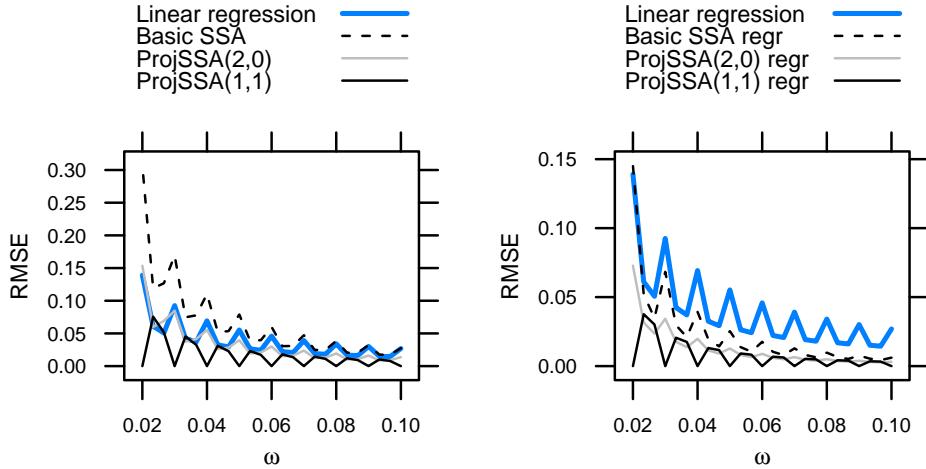


Рис. 2.3: Зависимость RMSE оценок линейного тренда от частоты периодической компоненты, $\phi = 0$.

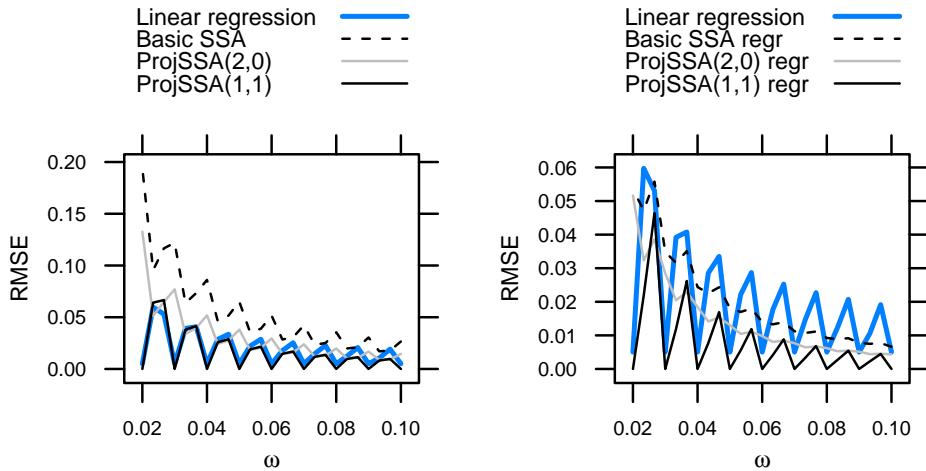


Рис. 2.4: Зависимость RMSE оценок линейного тренда от частоты периодической компоненты, $\phi = \pi/2$.

Заметим, что нулевые значения RMSE для ProjSSA(1,1) для частот $\omega = 0.01k$ объясняются теорией, поскольку тогда $L\omega$ и $K\omega$ являются целыми числами. Ошибки для ProjSSA(2,0) лежат между ошибками для Basic SSA и ProjSSA(1,1). Интересно, что минимальные ошибки для Basic SSA достигаются для средних точек, когда $L\omega + 0.5$ и $K\omega + 0.5$ — целые числа.

Замечание 5. Точность отделения периодической компоненты от тренда зависит от того, (a) является ли длина окна L малой или близкой к половине длины временного ряда и (b) является ли L кратной периоду периодической компоненты [1, Раздел 6.1]. Точность

более чувствительна к кратности (см. раздел 3.4). Поэтому, взяв разные периоды, мы проверяем стабильность сравнения в зависимости от выбора длины окна.

Кубический тренд и синусоида. Рассмотрим более сложный случай кубического тренда $t_n = 0.0001n^3$, $\sigma = 0$. Поскольку нет точной разделимости для любого выбора параметров, результаты непредсказуемы. Рис. 2.5 (слева) и 2.6 (справа) содержат значения RMSE для Basic SSA с восстановлением по ET1–4, ProjSSA(4,0), ProjSSA(2,2) с $L = 100$ и для кубической регрессии. Видно, что ProjSSA(2,2) является лучшим методом для $\phi = 0$, а для $\phi = \pi/2$ он просто сравним с линейной регрессией. Заметим, что здесь лучшие параметры для ProjSSA(2,2) не соответствуют случаю, когда $L\omega$ и $K\omega$ — целые числа. Кубическая аппроксимация по методу наименьших квадратов восстановленного тренда снова улучшает оценки (Рис. 2.5 (справа) и 2.6 (справа)).

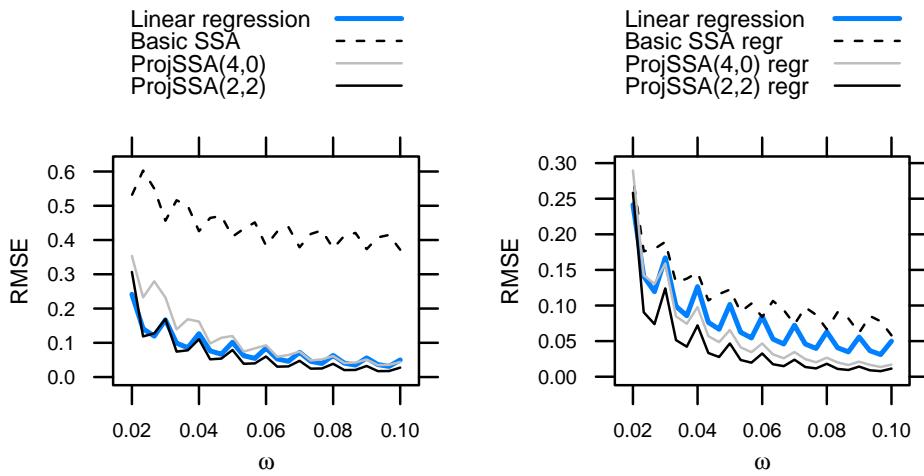


Рис. 2.5: Зависимость RMSE оценок кубического тренда от частоты периодической компоненты, $\phi = 0$.

Basic SSA не работает для выбранных параметров из-за отсутствия сильной разделимости: четвертая компонента тренда имеет вклад, сравнимый с вкладом периодических компонент, что приводит к их смешиванию.

Заметим, что модификация Iterative O-SSA, описанная в разделе 2.2, может быть использована для получения сильной точной разделимости для рассматриваемых бесшумных примеров. Однако, мы не включаем эту модификацию в сравнение, поскольку итеративный O-SSA не способен удалить шум и должен применяться после удаления шума вложенным способом, в то время как сравниваемые методы способны извлечь тренд без предварительного удаления шума.

Линейный тренд и шум. Для данных, которые удовлетворяют модели линейной регрессии с белым гауссовским шумом, то есть для амплитуды A равной нулю, мы берем $\sigma = 1$ и используем $M = 1000$. Как и ожидалось, для оценки регрессии достигается наименьшая ошибка 0,10. Однако, RMSE оценки ProjSSA(1,1), равная 0,12, очень близка к 0,10. Ошибка

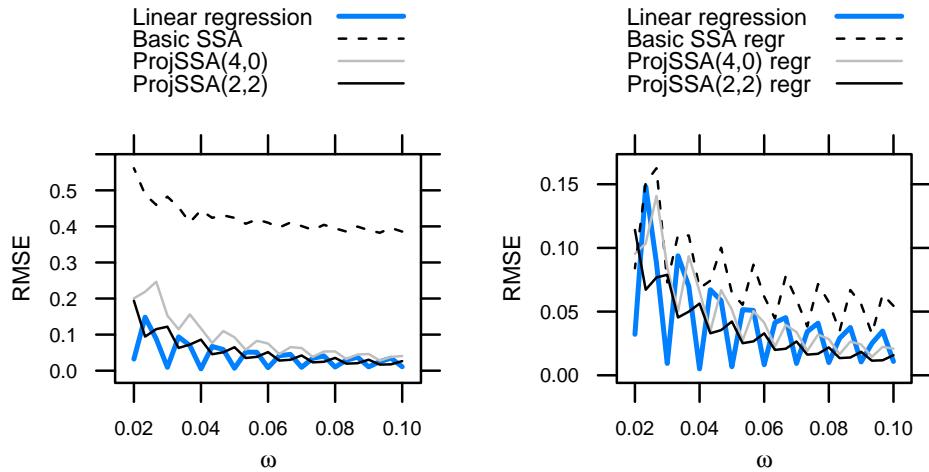


Рис. 2.6: Зависимость RMSE оценок кубического тренда от частоты периодической компоненты, $\phi = \pi/2$.

Basic SSA равна 0,17. Применение линейной регрессии к результатам восстановления SSA улучшает оценки SSA. RMSE для ProjSSA(1,1) и Basic SSA становятся равными 0,115 и 0,104 соответственно.

Мы не показываем ошибки оценок тренда, когда ряд имеет как периодическую составляющую, так и шум, поскольку результат сравнения (порядок значений ошибок) является промежуточным между случаями зашумленного тренда без периодической составляющей и тренда без шума с добавленной периодичностью. Общий вывод таков: чтобы сохранить преимущество SSA с проекцией, стандартное отклонение шума σ должно быть значительно меньше, чем амплитуда A периодической составляющей.

2.1.4. Обзор результатов

Рассмотренные применения SSA с проекцией были связаны с извлечением полиномиального тренда, поскольку его траекторное пространство определяется только степенью многочлена.

Мы показали на примере ‘Gasoline’, что подход линейной регрессии может быть неадекватным для коротких рядов и больших колебаний, по сравнению с ProjSSA(1,1). Сравнение различных версий SSA, примененных к данным ‘co2’, показывает, что даже если модель компоненты ряда используемая для проекций, неверна, непараметрическая часть разложения SSA с проекцией может исправить смещение.

Для лучшего понимания разницы между SSA с проекцией и подходом линейной регрессии, было проведено численное исследование. Во-первых, оказалось, что если мы извлекаем полиномиальный тренд с помощью SSA с проекцией, то полиномиальная аппроксимация наименьших квадратов восстановленного тренда может значительно повысить точность.

Второй обнаруженный эффект связан с влиянием формы остатка на точность оценки.

Соответственно, в рассматриваемом примере мы меняли фазу синусоиды. Оценки SSA слабо зависят от фазы, в то время как оценки тренда по линейной регрессии демонстрируют значительную зависимость.

Численные эксперименты подтверждают, что для линейного тренда и синусоидального остатка ProjSSA(1,1) более точен, чем оценка линейной регрессии. Для зашумленного линейного тренда, когда модель линейной регрессии верна, оценка линейной регрессии немноголибо точнее, чем SSA. Таким образом, можно сформулировать условия, при которых SSA с двойной проекцией может быть рекомендована к использованию: ряд имеет линейный или полиномиальный тренд (степень многочлена невелика) и регулярные колебания значительно превышают уровень шума.

Отметим, что еще одним применением ProjSSA является использование структуры вспомогательного ряда для улучшения восстановления и прогнозирования основного временного ряда.

2.2. Iterative Oblique SSA

В этом разделе мы расширяем понятие ортогональности, следуя [9], что позволяет ослабить условия слабой разделимости.

2.2.1. Oblique SVD и SSA

В этом разделе мы рассматриваем технику косоугольных (oblique) разложений, которая помогает расширить алгоритмы SSA и улучшить разделимость.

Ортогональность подрядов может быть сильным ограничением на разделяемые ряды. Однако, если мы рассмотрим ортогональность относительно нестандартного евклидового скалярного произведения, условия разделимости могут быть значительно ослаблены. В результате появился первый метод, названный Oblique SSA (O-SSA), в котором шаг SVD выполняется в неортогональной системе координат.

Скалярные произведения и связанные с ними разложения матриц

Здесь мы предоставляем необходимую информацию о разложениях матриц относительно заданного скалярного произведения в пространствах строк и столбцов (см., например, [116, Th.3]), называемые в [117] Restricted SVD (RSVD).

Скалярные произведения Обычно ортогональность векторов в \mathbb{R}^M рассматривается обычным образом: X_1 и X_2 в \mathbb{R}^M ортогональны, если их евклидово скалярное произведение равно 0, т.е. $(X_1, X_2)_M = 0$, где $(\cdot, \cdot)_M$ — стандартное скалярное произведение в \mathbb{R}^M . Иногда мы будем опускать размерность в обозначениях, если это ясно из контекста. Хорошо известно, что

любое скалярное произведение в \mathbb{R}^M может быть определено как $\langle X_1, X_2 \rangle_{\mathbf{A}} = X_1^T \mathbf{A} X_2$ для симметричной положительно-определенной матрицы \mathbf{A} . Для любого $\mathbf{O}_{\mathbf{A}}$ такого, что $\mathbf{O}_{\mathbf{A}}^T \mathbf{O}_{\mathbf{A}} = \mathbf{A}$ имеем $\langle X_1, X_2 \rangle_{\mathbf{A}} = (\mathbf{O}_{\mathbf{A}} X_1, \mathbf{O}_{\mathbf{A}} X_2)_M$. Очевидно, что $\mathbf{O}_{\mathbf{A}}$ определено с точностью до умножения на ортогональную матрицу.

Скалярное произведение дает понятие ортогональности. Мы будем говорить, что два вектора \mathbf{A} -ортогональны, если $\langle X_1, X_2 \rangle_{\mathbf{A}} = 0$.

Пусть матрица \mathbf{A} симметричная неотрицательно-определенная, $\text{rank } \mathbf{A} = r$. Тогда \mathbf{A} может быть разложена как $\mathbf{A} = \mathbf{O}_{\mathbf{A}}^T \mathbf{O}_{\mathbf{A}}$ с $\mathbf{O}_{\mathbf{A}} \in \mathcal{M}_{r,M}$ и $\langle X_1, X_2 \rangle_{\mathbf{A}} = (\mathbf{O}_{\mathbf{A}} X_1, \mathbf{O}_{\mathbf{A}} X_2)_r$. Заметим, что пространство строк $\mathbf{O}_{\mathbf{A}}$ одинаково при любом выборе $\mathbf{O}_{\mathbf{A}}$ и совпадает с пространством столбцов \mathbf{A} . Если матрица \mathbf{A} не является положительно определенной, то мы получим вырожденное скалярное произведение, то есть, если $\langle X, X \rangle_{\mathbf{A}} = 0$, то не обязательно, что $X = 0_M$. Однако для векторов, принадлежащих пространству столбцов \mathbf{A} , равенство $\langle X, X \rangle_{\mathbf{A}} = 0$ дает $X = 0_M$. Таким образом, если мы рассматриваем скалярное произведение, порожденное матрицей неполного ранга \mathbf{A} , то мы должны рассматривать его только на пространстве столбцов \mathbf{A} . В частности, мы можем корректно определить \mathbf{A} -ортогональность векторов из пространства столбцов \mathbf{A} .

Следующее очевидное предложение показывает, что любой базис можно считать \mathbf{A} -ортонормальным при некотором выборе $\mathbf{O}_{\mathbf{A}}$.

Предложение 5. Пусть P_1, \dots, P_r — множество линейно независимых векторов в \mathbb{R}^M . Тогда P_1, \dots, P_r являются \mathbf{A} -ортонормированными для $\mathbf{O}_{\mathbf{A}} = \mathbf{P}^\dagger$, где $\mathbf{P} = [P_1 : \dots : P_r]$.

Заметим, что пространство столбцов \mathbf{P} совпадает с пространством строк $\mathbf{O}_{\mathbf{A}}$. Мы называем матрицу $\mathbf{O}_{\mathbf{A}}$, которая делает вектора из множества P_1, \dots, P_r \mathbf{A} -ортонормированными ортонормирующими матрицей этого множества. Конечно, ортонормирующая матрица не является однозначно определенной.

Приведем элементарный пример. Пусть $X = (1, 2)^T$ и $Y = (1, 1)^T$. Конечно, эти векторы не ортогональны в обычном смысле: $(X, Y) = 3$. Однако, если мы определим

$$\mathbf{A} = \begin{pmatrix} 5 & -3 \\ -3 & 2 \end{pmatrix}, \quad (2.5)$$

тогда $\langle X, Y \rangle_{\mathbf{A}} = (\mathbf{A}X, Y) = 0$ и $(\mathbf{O}_{\mathbf{A}} X, \mathbf{O}_{\mathbf{A}} Y) = 0$ для любого $\mathbf{O}_{\mathbf{A}}$ такого, что $\mathbf{O}_{\mathbf{A}}^T \mathbf{O}_{\mathbf{A}} = \mathbf{A}$, например.

$$\mathbf{O}_{\mathbf{A}} = \begin{pmatrix} 1 & -1 \\ -2 & 1 \end{pmatrix}.$$

Это означает, что $\{X, Y\}$ является ортогональным базисом относительно \mathbf{A} скалярного произведения $\langle \cdot, \cdot \rangle_{\mathbf{A}}$ а $\mathbf{O}_{\mathbf{A}}$ соответствует ортогоанализирующему отображению. Матрица \mathbf{A} может быть выбрана такой, что X и Y имеют любую \mathbf{A} -норму. Выбор (2.5) соответствует \mathbf{A} -ортонормированности.

Косоугольные разложения Рассмотрим минимальные разложения матрицы $\mathbf{Y} \in \mathcal{M}_{L,K}$ ранга r в виде

$$\mathbf{Y} = \sum_{i=1}^r \sigma_i P_i Q_i^T, \quad (2.6)$$

где $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > 0$, $\{P_i\}_{i=1}^r$ и $\{Q_i\}_{i=1}^r$ линейно независимы (поэтому, $\{P_i\}_{i=1}^r$ — базис пространства столбцов \mathbf{Y} , $\{Q_i\}_{i=1}^r$ — базис пространства строк \mathbf{Y}). Удобно записать (2.6) в матричной форме: $\mathbf{Y} = \mathbf{P}\Sigma\mathbf{Q}^T$, где $\mathbf{P} = [P_1 : \dots : P_r]$, $\mathbf{Q} = [Q_1 : \dots : Q_r]$ и $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_r)$.

Предложение 6. Пусть \mathbf{O}_L — ортонормирующая матрица множества векторов $\{P_i\}_{i=1}^r$ и \mathbf{O}_R — ортонормирующая матрица множества $\{Q_i\}_{i=1}^r$. Тогда

$$\mathbf{O}_L \mathbf{Y} \mathbf{O}_R^T = \sum_{i=1}^r \sigma_i (\mathbf{O}_L P_i)(\mathbf{O}_R Q_i)^T \quad (2.7)$$

— это SVD матрицы $\mathbf{O}_L \mathbf{Y} \mathbf{O}_R^T \in \mathcal{M}_{r,r}$ с левыми сингулярными векторами $\mathbf{O}_L P_i \in \mathbb{R}^r$ и правыми сингулярными векторами $\mathbf{O}_R Q_i \in \mathbb{R}^r$.

Это предложение следует из того, что любое би-ортогональное разложение является SVD.

Определение 5. Если пространство столбцов \mathbf{L} содержит пространство столбцов \mathbf{Y} , а пространство столбцов \mathbf{R} содержит пространство строк \mathbf{Y} , то такую пару (\mathbf{L}, \mathbf{R}) мы будем называть согласованной с матрицей \mathbf{Y} .

Определение 6. Для (\mathbf{L}, \mathbf{R}) , согласованных с \mathbf{Y} , мы говорим, что (2.6) является (\mathbf{L}, \mathbf{R}) -SVD, если система $\{P_i\}_{i=1}^r$ является \mathbf{L} -ортонормированной и система $\{Q_i\}_{i=1}^r$ является \mathbf{R} -ортонормированной.

В матричной постановке задачи [117] (\mathbf{L}, \mathbf{R}) -SVD называется Restricted SVD матрицы \mathbf{Y} относительно (\mathbf{L}, \mathbf{R}) .

Из определения 6 следует, что (2.7) является SVD тогда и только тогда, когда (2.6) является (\mathbf{L}, \mathbf{R}) -SVD, где $\mathbf{L} = \mathbf{O}_L^T \mathbf{O}_L$ и $\mathbf{R} = \mathbf{O}_R^T \mathbf{O}_R$, \mathbf{O}_L и \mathbf{O}_R являются ортонормирующими.

Предложение 6 гласит, что любое минимальное разложение в сумму матриц ранга 1 в виде (2.6) является (\mathbf{L}, \mathbf{R}) -SVD для некоторых матриц \mathbf{L} и \mathbf{R} .

Предложение 7. Пусть

$$\mathbf{O}_L \mathbf{Y} \mathbf{O}_R^T = \sum_{i=1}^r \sqrt{\lambda_i} U_i V_i^T \quad (2.8)$$

— обычный SVD матрицы $\mathbf{O}_L \mathbf{Y} \mathbf{O}_R^T$. Тогда разложение (2.6) с $\sigma_i = \sqrt{\lambda_i}$, $P_i = \mathbf{O}_L^\dagger U_i$ и $Q_i = \mathbf{O}_R^\dagger V_i$ является (\mathbf{L}, \mathbf{R}) -SVD.

Предложение 7 вытекает из предложения 6 и дает метод вычисления (\mathbf{L}, \mathbf{R}) -SVD (см. Алгоритм 2.3).

Покажем, как можно изменить набор σ_i в (\mathbf{L}, \mathbf{R}) -SVD (2.6) без изменения направлений P_i и Q_i , то есть, $P_i/\|P_i\|$ и $Q_i/\|Q_i\|$.

Предложение 8. Пусть (2.6) — (\mathbf{L}, \mathbf{R}) -SVD с $\mathbf{O}_L = \mathbf{P}^\dagger$ и $\mathbf{O}_R = \mathbf{Q}^\dagger$. Тогда

$$\mathbf{Y} = \sum_{i=1}^r \tilde{\sigma}_i \tilde{P}_i \tilde{Q}_i^T, \quad (2.9)$$

где $\tilde{\sigma}_i = \sigma_i/(\mu_i v_i)$, $\tilde{P}_i = \mu_i P_i$ и $\tilde{Q}_i = v_i Q_i$ для некоторых положительных v_i и μ_i , является (после перестановки согласно упорядоченности $\tilde{\sigma}_i$) $(\tilde{\mathbf{L}}, \tilde{\mathbf{R}})$ -SVD с $\mathbf{O}_{\tilde{L}} = \tilde{\mathbf{P}}^\dagger$ и $\mathbf{O}_{\tilde{R}} = \tilde{\mathbf{Q}}^\dagger$.

Особую важность представляет случай односторонних неортогональных разложений, когда одна из матриц, \mathbf{R} или \mathbf{L} , идентична. В [117] показано, что тогда Restricted SVD является Quotient SVD (часто называемым обобщенным SVD [118]). Если \mathbf{L} — единичная матрица, то P_i , $i = 1, \dots, r$, являются ортонормированными в обычном смысле и образуют ортонормированный базис пространства столбцов \mathbf{Y} . Если \mathbf{R} — единичная матрица, то Q_i , $i = 1, \dots, r$, являются ортонормированными и составляют ортонормированный базис пространства строк.

Скалярные произведения матриц и аппроксимации Пусть $\mathbf{X}, \mathbf{Y} \in \mathcal{M}_{L,K}$, (\mathbf{L}, \mathbf{R}) согласованы с \mathbf{X} и \mathbf{Y} , $\mathbf{L} = \mathbf{O}_L^T \mathbf{O}_L$ и $\mathbf{R} = \mathbf{O}_R^T \mathbf{O}_R$.

Определим индуцированное скалярное произведение Фробениуса как

$$\langle \mathbf{X}, \mathbf{Y} \rangle_{F,(\mathbf{L},\mathbf{R})} = \langle \mathbf{O}_L \mathbf{X} \mathbf{O}_R^T, \mathbf{O}_L \mathbf{Y} \mathbf{O}_R^T \rangle_F.$$

Заметим, что определение не зависит от выбора \mathbf{O}_L и \mathbf{O}_R , поскольку $\langle \mathbf{C}, \mathbf{D} \rangle_F = \text{tr}(\mathbf{C}^T \mathbf{D}) = \text{tr}(\mathbf{C} \mathbf{D}^T)$.

Для двух матриц \mathbf{C} и \mathbf{D} мы говорим, что они

1. (\mathbf{L}, \mathbf{R}) F-ортогональны, если $\langle \mathbf{C}, \mathbf{D} \rangle_{F,(\mathbf{L},\mathbf{R})} = 0$,
2. (\mathbf{L}, \mathbf{R}) левоортогональны, если $\mathbf{C}^T \mathbf{L} \mathbf{D} = \mathbf{0}$,
3. (\mathbf{L}, \mathbf{R}) правоортогональны, если $\mathbf{C} \mathbf{R} \mathbf{D}^T = \mathbf{0}$,
4. (\mathbf{L}, \mathbf{R}) биортогональны, если левая и правая ортогональности имеют место.

Левая или правая ортогональность является достаточным условием для F-ортогональности. Матричные компоненты (\mathbf{L}, \mathbf{R}) -SVD являются (\mathbf{L}, \mathbf{R}) -биортогональными и поэтому (\mathbf{L}, \mathbf{R}) F-ортогональными.

Мера (\mathbf{L}, \mathbf{R}) -ортогональности равна

$$\rho_{(\mathbf{L},\mathbf{R})}(\mathbf{X}, \mathbf{Y}) = \frac{\langle \mathbf{X}, \mathbf{Y} \rangle_{F,(\mathbf{L},\mathbf{R})}}{\|\mathbf{X}\|_{F,(\mathbf{L},\mathbf{R})} \|\mathbf{Y}\|_{F,(\mathbf{L},\mathbf{R})}}. \quad (2.10)$$

Пусть $\mathbf{X} = \sum_i \mathbf{X}_i$, где $\mathbf{X}_i = \sigma_i P_i Q_i^T$, является (\mathbf{L}, \mathbf{R}) -SVD. Тогда $\|\mathbf{X}_i\|_{F,(\mathbf{L},\mathbf{R})} = \sigma_i$ и $\|\mathbf{X}\|_{F,(\mathbf{L},\mathbf{R})}^2 = \sum_i \sigma_i^2$. Тогда (\mathbf{L}, \mathbf{R}) -вклад слагаемого \mathbf{X}_k равен $\sigma_k^2 / \sum_i \sigma_i^2$.

Следующее предложение следует из представления скалярного произведения Фробениуса через след матричного умножения.

Предложение 9. *Если \mathbf{X} и \mathbf{Y} являются (\mathbf{L}, \mathbf{R}) -левоортогональными, то \mathbf{X} и \mathbf{Y} являются $(\mathbf{L}, \tilde{\mathbf{R}})$ F-ортогональными для любого $\tilde{\mathbf{R}}$.*

Следствие 3. *Пусть \mathbf{L} – единичная матрица, а \mathbf{X} и \mathbf{Y} – (\mathbf{L}, \mathbf{R}) -левоортогональны для некоторой матрицы \mathbf{R} . Тогда обычная F-ортогональность \mathbf{X} и \mathbf{Y} имеет место и $\|\mathbf{X} + \mathbf{Y}\|_F^2 = \|\mathbf{X}\|_F^2 + \|\mathbf{Y}\|_F^2$.*

Следствие 3 показывает, что если в пространстве строк или в пространстве столбцов матриц задано обычное скалярное произведение, то есть там векторы ортогональны в обычном смысле, то можно считать, что есть обычная F-ортогональность и поэтому F-норму и F-скалярное произведение можно использовать для измерения точности аппроксимации и ортогональности компонент.

Замечание 6. *Введенные определения и утверждения уместны, если \mathbf{L} и \mathbf{R} согласованы с матрицами \mathbf{X} и \mathbf{Y} (см. определение 5). В противном случае, например, величина (2.10) может быть формально вычислена, но эта мера будет отражать только корреляцию между проекциями столбцов и строк \mathbf{X} и \mathbf{Y} на пространства строк \mathbf{L} и \mathbf{R} соответственно.*

Заметим, что обычная норма Фробениуса является интерпретируемой характеристикой аппроксимации, тогда как норма, основанная на $\langle \cdot, \cdot \rangle_{F, (\mathbf{L}, \mathbf{R})}$, интерпретируется гораздо хуже, поскольку она эквивалентна аппроксимации по норме Фробениуса матрицы $\mathbf{O}_{\mathbf{L}} \mathbf{X} \mathbf{O}_{\mathbf{R}}^T$.

Oblique SSA

Хотя многие интерпретируемые компоненты ряда, такие как тренд (медленно меняющийся компонент) и сезонность, асимптотически ортогональны, для данной длины временного ряда ортогональность может быть не достигнута даже приближенно. Поэтому было бы полезно ослабить условие ортогональности. Предлагаемый подход заключается в использовании ортогональности, которая по-прежнему означает равенство скалярного произведения 0, но это неординарное скалярное произведение, которое приспособлено к компонентам временного ряда, которые мы хотим разделить.

Рассмотрим косоугольное скалярное произведение $\langle X_1, X_2 \rangle_{\mathbf{A}} = (\mathbf{A}X_1, X_2)$ с симметричной неотрицательно-определенной матрицей. Ниже, рассматривая $\langle X_1, X_2 \rangle_{\mathbf{A}}$, мы всегда будем считать, что векторы X_i , $i = 1, 2$, принадлежат пространству столбцов \mathbf{A} .

Для описания так называемого косоугольного (oblique) SSA будем рассматривать SVD матрицы \mathbf{X} , образованной двумя косоугольными базисами, \mathbf{L} -ортонормированным и \mathbf{R} -ортонормированным соответственно, в пространствах строк и столбцов (Определение 6).

Oblique SSA (O-SSA) – это модификация алгоритма Basic SSA, описанного в разделе 1.8, где шаг SVD заменен на (\mathbf{L}, \mathbf{R}) -SVD для некоторых матриц \mathbf{L} и \mathbf{R} , согласованных с \mathbf{X} (см.

определение 5). Мы будем использовать понятия, введенные в алгоритме Basic SSA, а также для его косоугольной модификации.

Предложение 7 дает алгоритм 2.3, который сводит (\mathbf{L}, \mathbf{R}) -SVD к обычной SVD.

АЛГОРИТМ 2.3: (\mathbf{L}, \mathbf{R}) -SVD.

Input: \mathbf{Y} ранга r , (\mathbf{L}, \mathbf{R}) согласован с \mathbf{Y} .

Output: SVD (\mathbf{L}, \mathbf{R}) в виде $\mathbf{Y} = \sum_{i=1}^r \sigma_i P_i Q_i^T$.

- 1: Вычислить \mathbf{O}_L и \mathbf{O}_R такие, что $\mathbf{O}_L^T \mathbf{O}_L = \mathbf{L}$ и $\mathbf{O}_R^T \mathbf{O}_R = \mathbf{R}$.
- 2: Вычислить $\mathbf{O}_L \mathbf{Y} \mathbf{O}_R^T$.
- 3: Найти обычное SVD-разложение $\mathbf{O}_L \mathbf{Y} \mathbf{O}_R^T = \sum_{i=1}^r \sqrt{\lambda_i} U_i V_i^T$.
- 4: $\sigma_i = \sqrt{\lambda_i}$, $P_i = \mathbf{O}_L^\dagger U_i$ и $Q_i = \mathbf{O}_R^\dagger V_i$. где \dagger обозначает псевдоинверсию.

Заметим, что если \mathbf{L} и \mathbf{R} — единичные матрицы, то Oblique SSA совпадает с Basic SSA, $\sigma_i = \sqrt{\lambda_i}$, $P_i = U_i$ и $Q_i = V_i$.

Разделимость Аналогично обычной разделимости, описанной в разделе 1.8.3, может быть введено понятие слабой и сильной (\mathbf{L}, \mathbf{R}) -разделимости. Опять же, пусть $\mathbf{X} = \mathbf{X}^{(1)} + \mathbf{X}^{(2)}$, \mathbf{X} — траекторная матрица ряда, $\mathbf{X}^{(m)}$ — траекторные матрицы компонентов ряда, $\mathbf{X}^{(m)} = \sum_{i=1}^{r_m} \sigma_{m,i} P_{m,i} Q_{m,i}^T$ — их (\mathbf{L}, \mathbf{R}) -SVDs, $m = 1, 2$. Мы предполагаем, что \mathbf{L} и \mathbf{R} согласованы с \mathbf{X} , $\mathbf{X}^{(1)}$ и $\mathbf{X}^{(2)}$.

Определение 7. Пусть L фиксировано. Два ряда $X_N^{(1)}$ и $X_N^{(2)}$ называются слабо (\mathbf{L}, \mathbf{R}) -разделимыми, если их столбцовые траекторные пространства \mathbf{L} -ортогональны, а также их строковые траекторные пространства \mathbf{R} -ортогональны, то есть

$$(\mathbf{X}^{(1)})^T \mathbf{L} \mathbf{X}^{(2)} = \mathbf{0}_{\mathbf{K}, \mathbf{K}}, \quad \mathbf{X}^{(1)} \mathbf{R} (\mathbf{X}^{(2)})^T = \mathbf{0}_{\mathbf{L}, \mathbf{L}}.$$

Определение 8. Два ряда $X_N^{(1)}$ и $X_N^{(2)}$ называются сильно (\mathbf{L}, \mathbf{R}) -разделимыми, если они слабо (\mathbf{L}, \mathbf{R}) -разделимы и $\sigma_{1,i} \neq \sigma_{2,j}$ для любых i и j .

(\mathbf{L}, \mathbf{R}) -разделимость двух компонентов ряда означает \mathbf{L} -ортогональность их подрядов длины L и \mathbf{R} -ортогональность их подрядов длины $K = N - L + 1$.

Следующая теорема показывает, что (\mathbf{L}, \mathbf{R}) -разделимость является в некотором смысле гораздо менее ограничительной, чем обычная.

Теорема 2. Пусть $\mathbf{X} = \mathbf{X}^{(1)} + \mathbf{X}^{(2)}$ — ряд длины N , L — длина окна и L -ранг \mathbf{X} равен r . Пусть $\mathbf{X}^{(m)}$ — ряд L -ранга r_m , $m = 1, 2$, $r_1 + r_2 = r$. Тогда существуют разделяющие матрицы $\mathbf{L} \in \mathcal{M}_{L,L}$ и $\mathbf{R} \in \mathcal{M}_{K,K}$ ранга r такие, что ряды $\mathbf{X}^{(1)}$ и $\mathbf{X}^{(2)}$ сильно (\mathbf{L}, \mathbf{R}) -разделимы.

Доказательство. Обозначим $\{P_i^{(m)}\}_{i=1}^{r_m}$ базис пространства столбцов $\mathbf{X}^{(m)}$ и $\{Q_i^{(m)}\}_{i=1}^{r_m}$ базис пространства строк $\mathbf{X}^{(m)}$, $m = 1, 2$; например, $P_i^{(m)} = P_{m,i} \in \mathbb{R}^L$, $Q_i^{(m)} = Q_{m,i} \in \mathbb{R}^K$. Определим

$$\mathbf{P} = [P_1^{(1)} : \dots : P_{r_1}^{(1)} : P_1^{(2)} : \dots : P_{r_2}^{(2)}],$$

$$\mathbf{Q} = [Q_1^{(1)} : \dots : Q_{r_1}^{(1)} : Q_1^{(2)} : \dots : Q_{r_2}^{(2)}].$$

По условиям теоремы, матрицы \mathbf{P} и \mathbf{Q} имеют полный ранг. Поскольку \mathbf{P}^\dagger и \mathbf{Q}^\dagger ортонормируют столбцы матриц \mathbf{P} и \mathbf{Q} (Предложение 5), траекторные матрицы $\mathbf{X}^{(1)}$ и $\mathbf{X}^{(2)}$ являются (\mathbf{L}, \mathbf{R}) -биортогональными для $\mathbf{L} = (\mathbf{P}^\dagger)^T \mathbf{P}^\dagger$ и $\mathbf{R} = (\mathbf{Q}^\dagger)^T \mathbf{Q}^\dagger$. Поэтому ряды $X^{(1)}$ и $X^{(2)}$ являются (\mathbf{L}, \mathbf{R}) -разделимыми.

Предложение 8 показывает, что можно изменить $\sigma_{m,i}$, сохраняя биортогональность; то есть, это предложение объясняет, как получить сильную разделимость, не испортив слабую.

□

Замечание 7. Рассмотрим два временных ряда, управляемых минимальными ЛРС порядков r_1 и r_2 , $r_1 + r_2 \leq \min(L, K)$. Условия теоремы 2 выполняются тогда и только тогда, когда множества характеристических корней ряда не пересекаются. Действительно, множества характеристических корней не пересекаются тогда и только тогда, когда пространства столбцов и строк L -траекторных матриц пересекаются только в $\{0\}$, то есть \mathbf{P} и \mathbf{Q} имеют полный ранг.

Замечание 8. Теорема 2 вместе с замечанием 7 показывает, что любые два временных ряда, управляемые ЛРС с различными характеристическими корнями, могут быть разделены некоторым (\mathbf{L}, \mathbf{R}) -SVD для достаточно больших рядов и длин окон.

Отметим, что Теорема 2 не является конструктивной, так как траекторные пространства разделенных рядов должны быть известны для точного разделения. Однако, мы можем попытаться оценить эти пространства и тем самым улучшить разделимость.

Меры косоугольной разделимости. Если Oblique SSA не разделяет компоненты точно, необходима мера качества разделимости. Мы можем рассмотреть аналог \mathbf{w} -корреляций, описанных в разделе 1.8.3, поскольку они определяются через скалярные произведения Фробениуса траекторных матриц и поэтому могут быть обобщены; см. раздел 2.2.1 для определения $\rho_{\mathbf{L}, \mathbf{R}}$ в (2.10). Определим (\mathbf{L}, \mathbf{R}) \mathbf{w} -корреляцию между восстановленными рядами $\tilde{\mathbf{X}}^{(1)}$ и $\tilde{\mathbf{X}}^{(2)}$ как $\rho_{(\mathbf{L}, \mathbf{R})}(\tilde{\mathbf{X}}^{(1)}, \tilde{\mathbf{X}}^{(2)})$. Заметим, что из-за диагонального усреднения, пространства столбцов и строк $\tilde{\mathbf{X}}^{(m)}$ не обязательно принадлежат пространствам столбцов \mathbf{L} и \mathbf{R} соответственно, то есть матрицы \mathbf{L} и \mathbf{R} могут быть не согласованы с $\tilde{\mathbf{X}}^{(m)}$, $m = 1, 2$. Поэтому $\rho_{\mathbf{L}, \mathbf{R}}$ учитывает только проекции столбцов и строк $\tilde{\mathbf{X}}^{(1)}$ и $\tilde{\mathbf{X}}^{(2)}$ на пространства столбцов \mathbf{L} и \mathbf{R} (замечание 6). Это означает, что $\rho_{\mathbf{L}, \mathbf{R}}$ может переоценить точность разделимости.

Для Oblique SSA, когда только одна из систем координат (левая или правая) является косоугольной, обычные \mathbf{w} -корреляции между рядами являются более подходящими мерами

разделимости, поскольку в случае точной косоугольной разделимости мы имеем ортогональные (по скалярному произведению Фробениуса) компоненты матрицы (следствие 3).

Другой важной мерой правильной разделимости является близость компонентов восстановленного ряда к временным рядам конечного ранга. Это может быть измерено вкладом ведущих $r_m = |I_m|$ собственных троек в SVD траекторной матрицы $\tilde{\mathbf{X}}^{(m)}$ m -ой компоненты восстановленного ряда $\tilde{\mathbf{X}}^{(m)}$. Если обозначить $\tilde{\lambda}_{m,i}$ собственные значения обычного SVD матрицы $\tilde{\mathbf{X}}^{(m)}$, то $\tau_{r_m}(\tilde{\mathbf{X}}^{(m)}) = 1 - \sum_{i=1}^{r_m} \tilde{\lambda}_{m,i} / \|\tilde{\mathbf{X}}^{(m)}\|^2$ отражает близость m -ого ряда к ряду ранга r_m .

Вложенный Oblique SSA В отличие от обычного SVD, SVD относительно неортогональных систем координат обеспечивает аппроксимацию не совсем подходящим способом. Поэтому Oblique SSA не может использоваться для выделения ведущих компонент, в частности, для выделения сигнала и для удаления шума.

Поэтому предлагается вложенный способ использования Oblique SSA. Этот подход в некоторой степени похож на факторный анализ, где факторное пространство может быть оценено с помощью анализа главных компонент, а затем из факторного пространства извлекаются интерпретируемые факторы.

Предположим, что Basic SSA может извлечь сигнал, но не может разделить компоненты сигнала. Например, пусть временной ряд состоит из зашумленной суммы двух синусоид. Тогда Basic SSA может удалить шум, но, вероятно, не сможет разделить эти синусоиды, если их частоты близки. Таким образом, Basic SSA используется для оценки подпространства суммы синусоид, а затем может быть использован другой метод для разделения самих синусоид. Выбор параметров для лучшего разделения подробно исследован в разделе 3.4.

Итак, применим Basic SSA с подходящими параметрами и пусть разложение матрицы $\mathbf{X} = \mathbf{X}_{I_1} + \dots + \mathbf{X}_{I_p}$ получено на этапе группировки в Basic SSA; каждая группа соответствует отдельной компоненте временного ряда. Пусть для уточненного разложения выбрана s -я группа $I = I_s$. Обозначим $\mathbf{Y} = \mathbf{X}_I$, $r = \text{rank } \mathbf{Y}$, $\mathbf{Y} = \mathcal{T}^{-1} \mathcal{H} \mathbf{Y}$ — ряд, полученный из \mathbf{Y} диагональным усреднением.

АЛГОРИТМ 2.4: Вложенный Oblique SSA.

Input: Матрица \mathbf{Y} , пара матриц (\mathbf{L}, \mathbf{R}) , которые согласованы с \mathbf{Y} (см. определение 5).

Output: Уточненное разложение ряда $\mathbf{Y} = \tilde{\mathbf{Y}}^{(1)} + \dots + \tilde{\mathbf{Y}}^{(l)}$.

- 1: Построить (\mathbf{L}, \mathbf{R}) -SVD матрицы \mathbf{Y} по алгоритму 2.3 в виде

$$\mathbf{Y} = \sum_{i=1}^r \sigma_i P_i Q_i^T.$$

- 2: Разбить множество $\{1, \dots, r\} = \bigsqcup_{m=1}^l J_m$ и выполнить группировку для получения уточненного разложения матрицы $\mathbf{Y} = \mathbf{Y}_{J_1} + \dots + \mathbf{Y}_{J_l}$.

3: Получить уточненное разложение ряда $\mathbf{Y} = \tilde{\mathbf{Y}}^{(1)} + \dots + \tilde{\mathbf{Y}}^{(l)}$, где $\tilde{\mathbf{Y}}^{(m)} = \mathcal{T}^{-1}\mathcal{H}\mathbf{Y}_{J_m}$.

Таким образом, после применения алгоритма 2.4 к группе I_s мы получаем следующее разложение ряда \mathbf{X} :

$$\mathbf{X} = \tilde{\mathbf{X}}^{(1)} + \dots + \tilde{\mathbf{X}}^{(p)}, \text{ где } \tilde{\mathbf{X}}^{(s)} = \tilde{\mathbf{Y}}^{(1)} + \dots + \tilde{\mathbf{Y}}^{(l)}.$$

Для простоты ниже мы рассмотрим случай $l = 2$.

При достаточно большой длине временного ряда и умеренном уровне шума, интерпретируемые компоненты, такие как тренды, колебания и шум, приближенно разделяются Basic SSA [1, Разделы 1.5 и 6.1]. Однако, условия приближенной разделимости могут быть ограничительными, особенно для коротких временных рядов.

Ортогональность подрядов, которая является основным условием разделимости в Basic SSA, см. раздел 1.8.3, может быть сильным ограничением на ряды, которые мы хотим разделить. Однако, если мы рассмотрим ортогональность по отношению к нестандартному евклидовому скалярному произведению, условия разделимости становятся значительно слабее. Этот подход дает метод, называемый Oblique SSA (O-SSA), при котором SVD выполняется в неортогональной системе координат на шаге разложения. Идея Oblique SSA похожа на идею предварительного отбеливания, которое часто используется в статистике в качестве предварительной обработки: если мы знаем ковариации между компонентами, то мы можем выполнить линейное преобразование и получить некоррелированные компоненты. Поскольку “ковариации” компонентов не известны заранее, используется итерационный метод, называемый Iterative Oblique SSA. Кроме того, этот метод способен изменить вклады компонентов определенным образом так, чтобы их сильная разделимость с большой вероятностью была улучшена.

2.2.2. Метод

Итеративный подход к O-SSA

Опишем итерационную версию вложенного O-SSA, которая представляет собой итерационный алгоритм для получения подходящих матриц \mathbf{L} и \mathbf{R} для (\mathbf{L}, \mathbf{R}) -SVD матрицы \mathbf{X}_I . Для правильного использования вложенных разложений мы должны предполагать, что матрица \mathbf{X}_I близка к траекторной матрице неполного ранга r .

Чтобы объяснить основной принцип метода, предположим, что $\mathbf{X}_I = \mathbf{Y}$ — траекторная матрица \mathbf{Y} . Пусть $\mathbf{Y} = \mathbf{Y}^{(1)} + \mathbf{Y}^{(2)}$ и траекторные матрицы \mathbf{Y}_1 и \mathbf{Y}_2 имеют ранги r_1 и r_2 , $r_1 + r_2 = r$. Тогда по теореме 2 существуют r -ранговые разделяющие матрицы \mathbf{L}^* , \mathbf{R}^* размеров $L \times L$ и $K \times K$ соответственно и разбиение $\{1, \dots, r\} = J_1 \sqcup J_2$ такое, что мы можем выполнить соответствующую группировку в $(\mathbf{L}^*, \mathbf{R}^*)$ -SVD и тем самым получить $\mathbf{Y}_{J_1} = \mathbf{Y}_1$ и $\mathbf{Y}_{J_2} = \mathbf{Y}_2$.

Разделяющие матрицы \mathbf{L}^* и \mathbf{R}^* неизвестны, так как они определяются неизвестными

траекторными пространствами $\mathbf{Y}^{(1)}$ и $\mathbf{Y}^{(2)}$. Поэтому мы хотим построить последовательность (\mathbf{L}, \mathbf{R}) -SVD разложений (2.6), которая в пределе дает требуемое разделяющее разложение.

Пусть мы имеем исходное $(\mathbf{L}^{(0)}, \mathbf{R}^{(0)})$ -SVD-разложение \mathbf{Y} и группируем его компоненты, чтобы получить некоторые начальные оценки $\tilde{\mathbf{Y}}^{(1,0)}$ и $\tilde{\mathbf{Y}}^{(2,0)}$ для $\mathbf{Y}^{(1)}$ и $\mathbf{Y}^{(2)}$. Тогда мы можем использовать траекторные пространства $\tilde{\mathbf{Y}}^{(1,0)}$ и $\tilde{\mathbf{Y}}^{(2,0)}$ для построения нового скалярного произведения, которое, как ожидается, будет ближе к разделяющему. Поэтому можно предположить, что $\tilde{\mathbf{Y}}^{(1,1)}$ и $\tilde{\mathbf{Y}}^{(2,1)}$ будут ближе к $\mathbf{Y}^{(1)}$ и $\mathbf{Y}^{(2)}$, и поэтому мы берем их траекторные пространства для построения нового скалярного произведения; и так далее. Конечно, если исходное разложение является сильно разделяющим, то мы получаем $\tilde{\mathbf{Y}}^{(m,1)} = \tilde{\mathbf{Y}}^{(m,0)} = \mathbf{Y}^{(m)}$, $m = 1, 2$.

Базовый итерационный алгоритм

Назовем итеративную версию вложенного Oblique SSA как *Iterative Oblique SSA* или *Iterative O-SSA*.

Как и раньше, мы рассматриваем вложенный O-SSA, входными данными которого является матрица $\mathbf{Y} = \mathbf{X}_I$ ранга r . Для Basic SSA и для вложенного O-SSA разбиение собственных троек для группировки производится после этапа разложения. Для итеративного O-SSA разбиение $I = \tilde{J}_1 \sqcup \tilde{J}_2$, $r_m = |\tilde{J}_m|$, должно быть задано заранее, так как итерации выполняются в автоматическом режиме. Конечно, выбор разбиения не делается вслепую, так как перед использованием вложенной версии мы имеем полное разложение, которое мы используем для выбора группы I и ее разбиения.

Iterative O-SSA состоит из повторного применения вложенного O-SSA с пересчетом $(\mathbf{L}^{(k)}, \mathbf{R}^{(k)})$. Как и любой итеративный алгоритм, итеративный O-SSA должен иметь начальные данные $(\mathbf{L}^{(0)}, \mathbf{R}^{(0)})$ и правило остановки. Стандартное правило остановки включает в себя максимальное количество итераций M и порог точности ϵ . В Iterative O-SSA алгоритм останавливается, если компоненты восстановленного ряда $\tilde{\mathbf{Y}}^{(m,k)}$, $m = 1, 2$, изменяются очень мало. Для функции $\rho(\cdot)$, определяющей норму вектора, итерации прекращаются при условии $\max(\rho(\tilde{\mathbf{Y}}^{(m,k)} - \tilde{\mathbf{Y}}^{(m,k-1)}), m = 1, 2) < \epsilon$.

Заметим, что поскольку согласованность (\mathbf{L}, \mathbf{R}) с матрицей \mathbf{Y} необходима для корректности (\mathbf{L}, \mathbf{R}) -SVD, начальные данные $(\mathbf{L}^{(0)}, \mathbf{R}^{(0)})$ также должны быть согласованными.

Замечание 9. Начальные матрицы $(\mathbf{L}^{(0)}, \mathbf{R}^{(0)})$ вместе с группировкой могут быть заданы так, что исходное разложение является частью SVD (1.12), заданного множеством индексов I и $I = \tilde{J}_1 \sqcup \tilde{J}_2$, где \tilde{J}_1 и \tilde{J}_2 выбраны на основе анализа компонентов в (1.12). Так как разложение (1.12) биортогонально, $\mathbf{L}^{(0)}$ и $\mathbf{R}^{(0)}$ – единичные матрицы. Удобно обозначить через J_1 и J_2 множества, состоящие из порядковых индексов элементов \tilde{J}_1 и \tilde{J}_2 в I . Таким образом, $\{1, \dots, r\} = J_1 \sqcup J_2$. Например, если $\tilde{J}_1 = \{11, 14\}$ и $\tilde{J}_2 = \{12, 18\}$, то $I = \{11, 12, 14, 18\}$, $J_1 = \{1, 3\}$ и $J_2 = \{2, 4\}$.

В завершение описания Iterative O-SSA мы представляем формальное описание итераций. Разделяющее разложение $\mathbf{Y} = \mathbf{Y}_1 + \mathbf{Y}_2$ должно удовлетворять следующим свойствам:

- (a) \mathbf{Y}_1 и \mathbf{Y}_2 — ганкелевы;
- (b) $\text{rank } \mathbf{Y}_1 = r_1$, $\text{rank } \mathbf{Y}_2 = r_2$;
- (c) пространства столбцов и строк матриц \mathbf{Y}_1 и \mathbf{Y}_2 лежат в пространстве столбцов и строк матрицы \mathbf{Y} ;
- (d) \mathbf{Y}_1 и \mathbf{Y}_2 являются (\mathbf{L}, \mathbf{R}) -биортогональными для $\mathbf{L} = \mathbf{L}^*$ и $\mathbf{R} = \mathbf{R}^*$.

Определим Π_{col} ортогональный оператор проектирования (по евклидовой нормы) на пространство столбцов \mathbf{Y} , Π_{row} оператор проектирования на пространство строк \mathbf{Y} . Вложенная группа является упорядоченным объединением $I = \tilde{J}_1 \sqcup \tilde{J}_2$, $r_m = |\tilde{J}_m|$, J_1 и J_2 определены в замечании 9; пара матриц $(\mathbf{L}^{(k-1)}, \mathbf{R}^{(k-1)})$ является входными данными для k -ой итерации.

Сформулируем шаги k -ой итерации.

- (A) Чтобы получить ганкелевы матрицы, мы выполняем ганкелизацию входного разложения $\tilde{\mathbf{Y}}_m = \Pi_{\mathcal{H}} \mathbf{Y}_{J_m}^{(k-1)}$, $m = 1, 2$.
- (B) Затем, чтобы получить низкоранговые приближения рангов r_1 и r_2 соответственно, мы строим обычные SVD $\tilde{\mathbf{Y}}_m = \sum_{i=1}^{d_m} \sqrt{\lambda_i^{(m)}} U_i^{(m)} (V_i^{(m)})^T$, $m = 1, 2$, и берем ведущие члены r_m .
- (C) Поскольку мы не должны выходить за пределы пространства столбцов входной матрицы \mathbf{Y} (мы рассматриваем вложенное разложение), мы находим проекции $\hat{U}_i^{(m)} = \Pi_{\text{col}} U_i^{(m)}$ и $\hat{V}_i^{(m)} = \Pi_{\text{col}} \Pi_{\text{row}} V_i^{(m)}$ для $i = 1, \dots, r_m$, $m = 1, 2$. Обозначим

$$\hat{\mathbf{U}}^{(m)} = [\hat{U}_1^{(m)} : \dots : \hat{U}_{r_m}^{(m)}], \quad \hat{\mathbf{V}}^{(m)} = [\hat{V}_1^{(m)} : \dots : \hat{V}_{r_m}^{(m)}].$$

Для корректности алгоритма предполагаем, что матрицы $\hat{\mathbf{U}}^{(m)}$ и $\hat{\mathbf{V}}^{(m)}$ имеют полный ранг; в противном случае алгоритм может не сработать.

- (D) Наконец, вычислим $\mathbf{L}^{(k)} = (\hat{\mathbf{U}}^\dagger)^T \hat{\mathbf{U}}^\dagger$ и $\mathbf{R}^{(k)} = (\hat{\mathbf{V}}^\dagger)^T \hat{\mathbf{V}}^\dagger$, где $\hat{\mathbf{U}} = [\hat{\mathbf{U}}^{(1)} : \hat{\mathbf{U}}^{(2)}]$ and $\hat{\mathbf{V}} = [\hat{\mathbf{V}}^{(1)} : \hat{\mathbf{V}}^{(2)}]$, для достижения $(\mathbf{L}^{(k)}, \mathbf{R}^{(k)})$ -биортогональности.

Сходимость $\tilde{\mathbf{Y}}^{(1,k)}$ и $\tilde{\mathbf{Y}}^{(2,k)}$ к правильным разложениям теоретически не доказана. Однако, глядя на схему построения, которая напоминает поопеременные проекции, мы ожидаем сходимости, по крайней мере, в типичных случаях. Численные эксперименты подтверждают сходимость в большинстве примеров. Отметим также, что итеративный O-SSA не изменяет разделяющее разложение; то есть, разделяющее разложение является неподвижной точкой алгоритма.

Модификация с сигма-коррекцией

Если начальная точка $(\mathbf{L}^{(0)}, \mathbf{R}^{(0)})$ для итераций находится недалеко от разделяющей пары $(\mathbf{L}^*, \mathbf{R}^*)$, то можно ожидать, что сходимость в приведенном выше алгоритме будет иметь место, так как мы близки к значению неподвижной точки и можно ожидать, что $\sigma_i^{(k)}$ в $(\mathbf{L}^{(k)}, \mathbf{R}^{(k)})$ -SVD $\mathbf{Y} = \sum_{i=1}^r \sigma_i^{(k)} P_i^{(k)} (Q_i^{(k)})^\top$ лишь слегка изменяются во время итераций. В общем случае, однако, возможное изменение порядка следования компонентов разложения между итерациями Iterative O-SSA может помешать сходимости. Случай $J_1 = \{1, \dots, r_1\}$, когда минимальное сингулярное число σ_{r_1} первого ряда сохраняется значительно больше, чем максимальное сингулярное число σ_{r_1+1} второго ряда, предотвращает переупорядочивание компонент и, следовательно, улучшит сходимость.

Опишем модификацию Iterative O-SSA, которая обеспечивает переупорядочивание компонентов, раздвигает их друг от друга и тем самым снимает проблему смешивания компонентов. В этой модификации вносится корректировка для вычисления $\widehat{\mathbf{U}}^{(2)}$ и $\widehat{\mathbf{V}}^{(2)}$ на шаге (C) итераций.

Выберем параметр $\kappa > 1$. Если $\lambda_{r_1}^{(1)} < \kappa^2 \lambda_1^{(2)}$ на шаге (C), то определим $\mu = \kappa \sqrt{\lambda_1^{(2)} / \lambda_{r_1}^{(1)}}$ и изменим $\widehat{\mathbf{U}}^{(2)} \leftarrow \sqrt{\mu} \widehat{\mathbf{U}}^{(2)}$, $\widehat{\mathbf{V}}^{(2)} \leftarrow \sqrt{\mu} \widehat{\mathbf{V}}^{(2)}$. Чтобы быть последовательными с переупорядочиванием, зададим $J_1 = \{1, \dots, r_1\}$, $J_2 = \{r_1 + 1, \dots, r\}$.

Заметим, что если $\lambda_{r_1}^{(1)} < \kappa^2 \lambda_1^{(2)}$, то приведенная выше корректировка меняет порядок следования компонентов матрицы компонентов в (2.11), так как они упорядочиваются на $\sigma_i^{(k)}$. Следовательно, мы вынуждены увеличить компоненты матрицы, соотносящиеся с первым компонентом ряда. О том, как работает сигма-коррекция, см. предложение 8.

Замечание 10. Процедура переупорядочивания осуществляется путем последовательной корректировки весов компонентов и поэтому зависит от нумерации компонентов.

Подводя итог, можно сказать, что описанная коррекция может помочь улучшить сходимость и обеспечить сильную разделимость компонентов в случае, когда имеет место только слабая разделимость.

2.2.3. Алгоритмы

Представим метод, описанный в разделе 2.2 в виде алгоритмов. Метод состоит из различных частей и поэтому мы описываем его в виде нескольких алгоритмов.

Итак, сформулируем алгоритм Iterative O-SSA. Обозначим $\mathbf{Y} = \mathbf{X}_I$, $r = \text{rank } \mathbf{Y}$, $\mathbf{Y} = \mathcal{T}^{-1} \circ \Pi_{\mathcal{H}}(\mathbf{Y})$ ряд, полученный из \mathbf{Y} путем диагонального усреднения.

Мы разделяем весь алгоритм на две части. Алгоритм 2.5 показывает общую схему итеративного O-SSA, но не показывает, как вычислить пару матриц $(\mathbf{L}^{(k)}, \mathbf{R}^{(k)})$ на каждой итерации. Алгоритм 2.6 восполняет этот пробел.

 АЛГОРИТМ 2.5: Iterative O-SSA

Input: Разложение L -траекторной матрицы $\mathbf{X} = \sum_{i=1}^d \sigma_i P_i Q_i^T$ ряда X ; непересекающиеся множества индексов \tilde{J}_1 и \tilde{J}_2 из $\{1, \dots, d\}$; граница точности ϵ ; функция ρ для вычисления точности; максимальное количество итераций M ; начальные матрицы $(\mathbf{L}^{(0)}, \mathbf{R}^{(0)})$ соответствующие $\mathbf{Y} = \mathbf{X}_I$. Множество $I = \{i_1, \dots, i_r\}$ определяется как $I = \tilde{J}_1 \sqcup \tilde{J}_2$, $r_m = |\tilde{J}_m|$, $r = |I| = r_1 + r_2$, множества J_1 и J_2 определены в замечании 9. Это разбиение дает разложения для матриц и рядов: $\mathbf{Y} = \mathbf{Y}_{J_1}^{(0)} + \mathbf{Y}_{J_2}^{(0)}$ и $\mathbf{Y} = \tilde{\mathbf{Y}}^{(1,0)} + \tilde{\mathbf{Y}}^{(2,0)}$.

Output: $\mathbf{Y} = \tilde{\mathbf{Y}}^{(1)} + \tilde{\mathbf{Y}}^{(2)}$.

- 1: Задать $k = 1$.
- 2: Вызвать алгоритм 2.6 для вычисления $(\mathbf{L}^{(k)}, \mathbf{R}^{(k)})$, согласованного с \mathbf{Y} .
- 3: Вычислить $(\mathbf{L}^{(k)}, \mathbf{R}^{(k)})$ -SVD матрицы \mathbf{Y} по алгоритму 2.3:

$$\mathbf{Y} = \sum_{i=1}^r \sigma_i^{(k)} P_i^{(k)} (Q_i^{(k)})^T = \mathbf{Y}_{J_1}^{(k)} + \mathbf{Y}_{J_2}^{(k)}. \quad (2.11)$$

- 4: Получить разложение ряда $\mathbf{Y} = \tilde{\mathbf{Y}}^{(1,k)} + \tilde{\mathbf{Y}}^{(2,k)}$, где $\tilde{\mathbf{Y}}^{(m,k)} = \mathcal{T}^{-1} \circ \Pi_{\mathcal{H}}(\mathbf{Y}_{J_m}^{(k)})$, $m = 1, 2$.
 - 5: Если $k \geq M$ или $\max(\rho(\tilde{\mathbf{Y}}^{(m,k)} - \tilde{\mathbf{Y}}^{(m,k-1)}), m = 1, 2) < \epsilon$, то $\tilde{\mathbf{Y}}^{(m)} \leftarrow \tilde{\mathbf{Y}}^{(m,k)}$, $m = 1, 2$, и СТОП; иначе $k \leftarrow k + 1$ и перейти к шагу 2.
-

Алгоритм 2.6 представляет саму итерацию, включая сигма-коррекцию, которая может быть полезна для достижения сильной разделимости.

 АЛГОРИТМ 2.6: Расчет $(\mathbf{L}^{(k)}, \mathbf{R}^{(k)})$

Input: Разбиение $\{1, \dots, r\} = J_1 \sqcup J_2$; $r_m = |J_m|$; пара матриц $(\mathbf{L}^{(k-1)}, \mathbf{R}^{(k-1)})$; параметр для сигма-коррекции $\varkappa > 1$ (если $\varkappa = 0$, то сигма-коррекция не выполняется).

Output: Пара матриц $(\mathbf{L}^{(k)}, \mathbf{R}^{(k)})$ для k -ой итерации.

- 1: Вычислить $\tilde{\mathbf{Y}}_m = \Pi_{\mathcal{H}} \mathbf{Y}_{J_m}^{(k-1)}$, $m = 1, 2$.
- 2: Построить обычные SVD:

$$\tilde{\mathbf{Y}}_m = \sum_{i=1}^{d_m} \sqrt{\lambda_i^{(m)}} U_i^{(m)} (V_i^{(m)})^T, \quad m = 1, 2,$$

(нам нужны только первые r_m члены).

- 3: Сигма-коррекция (если $\varkappa \neq 0$): Если $\lambda_{r_1}^{(1)} < \varkappa^2 \lambda_1^{(2)}$, то определить $\mu = \varkappa \sqrt{\lambda_1^{(2)} / \lambda_{r_1}^{(1)}}$ и изменить $\hat{\mathbf{U}}^{(2)} \leftarrow \sqrt{\mu} \hat{\mathbf{U}}^{(2)}$, $\hat{\mathbf{V}}^{(2)} \leftarrow \sqrt{\mu} \hat{\mathbf{V}}^{(2)}$. Ввиду перенормировки, зададим $J_1 = \{1, \dots, r_1\}$, $J_2 = \{r_1 + 1, \dots, r\}$.

- 4: Найти проекции $\widehat{U}_i^{(m)} = \Pi_{\text{col}} U_i^{(m)}$ и $\widehat{V}_i^{(m)} = \Pi_{\text{row}} V_i^{(m)}$ для $i = 1, \dots, r_m$, $m = 1, 2$. Обозначим $\widehat{\mathbf{U}}^{(m)} = [\widehat{U}_1^{(m)} : \dots : \widehat{U}_{r_m}^{(m)}]$, $\widehat{\mathbf{V}}^{(m)} = [\widehat{V}_1^{(m)} : \dots : \widehat{V}_{r_m}^{(m)}]$.
- 5: Вычислить $\mathbf{L}^{(k)} = (\widehat{\mathbf{U}}^\dagger)^T \widehat{\mathbf{U}}^\dagger$ и $\mathbf{R}^{(k)} = (\widehat{\mathbf{V}}^\dagger)^T \widehat{\mathbf{V}}^\dagger$, где $\widehat{\mathbf{U}} = [\widehat{\mathbf{U}}^{(1)} : \widehat{\mathbf{U}}^{(2)}]$ и $\widehat{\mathbf{V}} = [\widehat{\mathbf{V}}^{(1)} : \widehat{\mathbf{V}}^{(2)}]$.
-

Замечание 11. Алгоритм 2.5, который использует сигма-коррекцию, может изменять группы индексов. Новые группы в алгоритме 2.6 строятся таким образом, что J_1 и J_2 разбивают множество $\{1, \dots, r\}$. Новое разбиение I получается как $\tilde{J}_1 = \{i_k \in I : k \in J_1\}$ и $\tilde{J}_2 = \{i_k \in I : k \in J_2\}$.

Замечание 12. Алгоритм 2.5 описывает уточненное разложение матрицы \mathbf{X}_I . Однако, мы можем рассматривать Iterative O-SSA как алгоритм, в котором получается полное разложение траекторной матрицы \mathbf{X} исходного ряда X (в результате которого меняются компоненты из группы I). Результатом также будет полное разложение.

2.2.4. Пример. Разделимость синусоид с близкими частотами

Случай отсутствия шума Рассмотрим сумму двух синусоид $x_n = \sin(2\pi\omega_1 n) + A \sin(2\pi\omega_2 n)$, $n = 1, \dots, N$, $N = 150$, с близкими частотами $\omega_1 = 0.065$ и $\omega_2 = 0.06$ и неравными амплитудами, 1 и $A = 1.2$. Пусть длина окна $L = 70$. Поскольку синусоиды с такими близкими частотами далеко не ортогональны для рассматриваемых длин окна и ряда, Basic SSA не может их разделить, см. рис. 2.7 (вверху), где показан результат разложения Basic SSA.

Для разделения синусоид мы применяем алгоритм Iterative O-SSA (алгоритм 2.5) без сигма-коррекции, $\epsilon = 10^{-5}$ и две группы ET1–2 и ET3–4. Максимальное число M итераций было взято очень большим и поэтому не было достигнуто. Разложение после применения Iterative O-SSA изображена на рис. 2.7 (внизу).

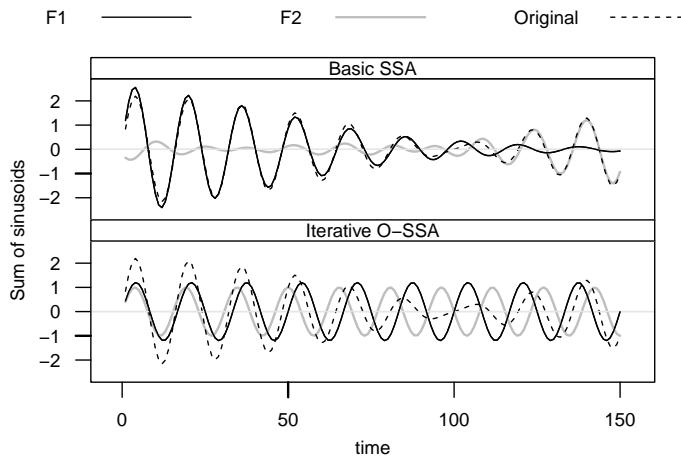


Рис. 2.7: Сумма двух синусоид с близкими частотами: разложение с помощью Basic SSA и Iterative O-SSA.

Давайте применим меры разделимости, описанные в разделе 2.2.1. Заметим, что обычные \mathbf{w} -корреляции не отражают качество разложения. Для начального разложения мы имеем 0.08. После итеративного O-SSA \mathbf{w} -корреляция становится равной -0.44 , в то время как (\mathbf{L}, \mathbf{R}) \mathbf{w} -корреляция почти равна 0. Последний результат подтверждает, что метод точно разделяет гармоники. Другой мерой истинности разложения является близость компонентов к рядам конечных рангов. Поскольку ранги должны быть равны числу компонент в выбранных группах, мы можем вычислить долю соответствующего числа ведущих компонент в их SVD-разложениях. Средняя доля $0.5(\tau_{r_1}(X^{(1)}) + \tau_{r_2}(X^{(2)}))$ изменяется с 0.06 почти до 0.

Зафиксируем $\omega_2 = 0.06$. Тогда для $\omega_1 = 0.065$ алгоритм останавливается после 113 итераций, для $\omega_1 = 0.07$ число итераций равно 26, а для $\omega_1 = 0.08$ — всего 6; см. толстую серую линию на рис. 2.9 (вверху).

Обратим внимание, что нам не нужно использовать сигма-коррекцию, так как синусоиды имеют разные амплитуды.

Если рассматривать равные амплитуды с $A = 1$ и принять $\varkappa = 2$ (алгоритмы 2.5 и 2.6), то Iterative O-SSA сходится к истинному решению даже при $\omega_2 = 0.065$ (191 итерация).

Вложенная разделимость при наличии шума Давайте добавим шум к сумме двух синусоид и возьмем $x_n = \sin(2\pi\omega_1 n) + A \sin(2\pi\omega_2 n) + \delta \varepsilon_n$ с близкими частотами $\omega_1 = 0.07$ и $\omega_2 = 0.06$ и неравными амплитудами, 1 и $A = 1.2$. Здесь ε_n — белый гауссовский шум с дисперсией 1, $\delta = 1$. Пусть снова $N = 150$, $L = 70$.

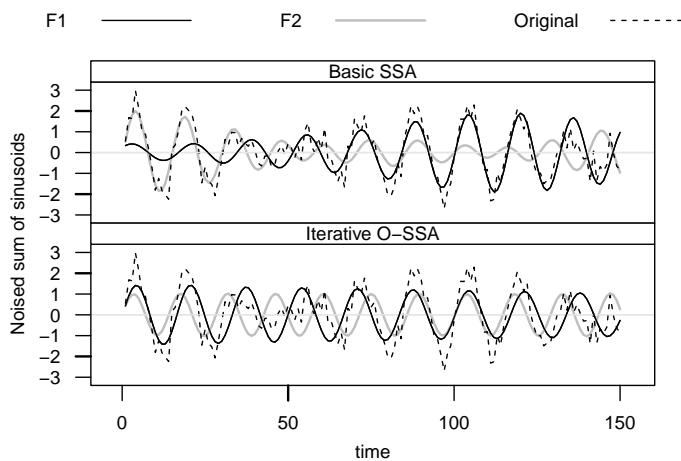


Рис. 2.8: Зашумленная сумма двух синусоид с близкими частотами: разложение с помощью Basic SSA и Iterative O-SSA.

Basic SSA хорошо отделяет синусоиды от шума, но не может отделить сами синусоиды. Таким образом, необходимо использовать Iterative O-SSA, применяемый к оцененному подпространству сигнала. Мы используем сигма-коррекцию с $\varkappa = 2$, так как разница между амплитудами 1 и 1.2 оказывается малой для сильной разделимости при наличии шума. Как и раньше, мы установили начальную группировку ET1–2 и ET3–4.

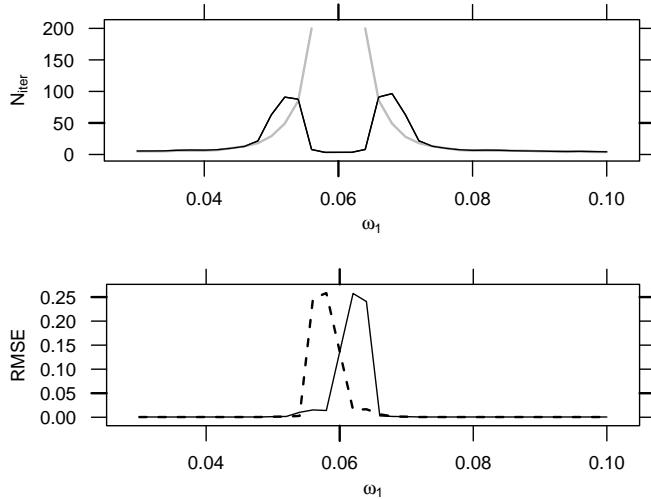


Рис. 2.9: Зависимость числа итераций (вверху) и ошибки RMSE оценок частоты (внизу) от ω_1 для $\omega_2 = 0.6$.

Разложение с помощью Basic SSA вверху и с помощью Iterative O-SSA внизу изображено на рис. 2.8. Число итераций равно 32, что лишь немного больше, чем 26 в случае отсутствия шума.

Исследуем зависимость числа итераций от ω_1 при фиксированном $\omega_2 = 0.06$. Мы изменяем ω_1 от 0.03 до 0.059 и от 0.061 до 0.1. На рис. 2.9 (сверху) показано количество итераций для сигнала без шума (толстая серая линия) и оцененное среднее число итераций для зашумленного сигнала (тонкая черная линия); число повторений равно 1000, вычислены 5%-винсORIZED оценки средних. Обращаем внимание, что число итераций было ограничено 200, при этом для чистого сигнала сходимость имела место для каждого ω_1 из рассматриваемого набора. Удивительно малое число итераций для зашумленного сигнала и близких частот объясняется сходимостью к неверному пределу, см. рис. 2.9 (внизу) со среднеквадратичными ошибками LS-ESPRIT оценок для ω_1 и ω_2 , основанных на подпространствах, охватываемых собственными векторами из ET1–2 и ET3–4 (см., например, [108] или [29, Section 2.8.2] для алгоритмов ESPRIT; сам алгоритм коротко описан в разделе 1.9.2). Поскольку мы используем вложенное разложение, шум незначительно влияет на точность восстановления для частот, которые сильно отличаются (ω_1 меньше 0.048 и больше 0.072).

2.3. Filter-adjusted O-SSA и SSA с производными

В этом разделе, следуя [9], мы описываем модификацию метода SSA, которая помогает преодолеть проблему отсутствия сильной разделимости при наличии слабой разделимости.

2.3.1. SSA с производными. Вариация для сильной разделимости

Напомним, что отсутствие сильной разделимости компонентов двух рядов обусловлено равенством сингулярных чисел в наборах сингулярных чисел, порожденных каждым из временных рядов. В свою очередь, сингулярные числа зависят от коэффициентов A_1 и A_2 перед компонентами ряда в сумме $A_1 s_n^{(1)} + A_2 s_n^{(2)}$. Вопрос в том, как изменить коэффициенты A_1 и A_2 в условиях неизвестных $s_n^{(1)}$ и $s_n^{(2)}$ так, чтобы сингулярные числа были различными.

Представляется, что наиболее естественным подходом является использование производной временного ряда для изменения коэффициентов, а не для изменения подпространств компонентов. Например, если $x_n = A \sin(2\pi\omega n + \phi)$, то $x'_n = 2\pi\omega A \cos(2\pi\omega n + \phi)$, то есть коэффициент $A' = 2\pi\omega A$. Если взять две синусоиды с разными частотами, то производная изменяет их амплитуды по-разному. Для $x_n = Ae^{\alpha n}$ производная также изменяет коэффициент перед экспонентой, так как $x'_n = \alpha A e^{\alpha n}$, и сохраняет коэффициент. Для большинства рядов конечного ранга подпространство производной совпадает с подпространством ряда. Исключением является полиномиальный ряд, когда производное подпространство является подмножеством начального подпространства.

Конечно, поскольку мы имеем дело с дискретным временем, мы рассматриваем $\varphi_n(X) = x_{n+1} - x_n$ вместо производной. Тем не менее, подход, основанный на взятии разностей, работает. Например, для ряда $X = X_N$ длины N с $x_n = A \sin(2\pi\omega n + \phi)$, мы получим ряд $\Phi_{N-1}(X) = (\varphi_1(X), \dots, \varphi_{N-1}(X))$ длины $N-1$ с $\varphi_n(X) = 2 \sin(\pi\omega) A \cos(2\pi\omega n + \pi\omega + \phi)$; для $x_n = Ae^{\alpha n}$, получаем $\varphi_n(X) = (e^\alpha - 1)Ae^{\alpha n}$.

Таким образом, мы можем объединить исходный ряд и его производную, чтобы уравновесить вклад компонентов и, следовательно, получить их сильную разделимость. Для синусоиды, чем меньше период, тем больше происходит увеличение амплитуды синусоиды после дифференцирования. Поэтому производная увеличивает вклад высоких частот. Этот эффект может увеличить уровень шумовой составляющей, если ряд испорчен шумом. Следовательно, следует производить вложенную версию реализации метода; в частности, шумовая составляющая должна быть предварительно удалена с помощью Basic SSA.

Замечание 13. *Подход с использованием производных (т.е., последовательных разностей) может быть естественным образом распространен на применение произвольной линейной фильтрации φ вместо взятия последовательных разностей. В данной работе мы имеем дело с производными, поскольку этот частный случай прост и имеет очень полезные приложения.*

В следующем разделе мы рассмотрим исходный ряд и его производную вместе как два ряда, регулируя вклад производной, а затем применим многомерную версию SSA. Затем этот подход трансформируется в специальную вложенную версию Oblique SSA, называемую DerivSSA.

SSA с производными как MSSA

Рассмотрим систему двух временных рядов $(X_N, \gamma\Phi_{N-1}(X))$ и применим многомерный SSA (MSSA). Алгоритм MSSA коротко упомянут в разделе 1.3.1 и подробно описан в главе 4.

В MSSA оператор встраивания \mathcal{T} переносит два временных ряда (X_{N_1}, Y_{N_2}) в сложенную L -траекторную матрицу $[X : Y]$, т.е., единственное отличие от Basic SSA состоит в построении оператора вложения \mathcal{T} .

Пусть $X_N = X_N^{(1)} + X_N^{(2)}$, где $X_N^{(1)}$ и $X_N^{(2)}$ имеют конечный ранг и приближенно разделимы. Поэтому их строковые и столбцовые траекторные пространства приближенно ортогональны. Тогда то же самое справедливо для $\Phi_{N-1}(X^{(1)})$ и $\Phi_{N-1}(X^{(2)})$ ввиду того, что их пространства столбцов являются подмножествами пространства столбцов рядов $X_N^{(1)}$ и $X_N^{(2)}$, а их пространства строк являются линейными оболочками векторов той же структуры, что и векторы, составляющие базисы строковых траекторных пространств рядов $X_N^{(1)}$ и $X_N^{(2)}$, за исключением того, что эти базисные векторы имеют длину $K - 1$, а не K . Поэтому приближенная ортогональность все еще сохраняется. Так как $\Phi_{N-1}(X) = \Phi_{N-1}(X^{(1)}) + \Phi_{N-1}(X^{(2)})$, MSSA, примененный к $(X_N, \gamma\Phi_{N-1}(X))$, приближенно разделит временные ряды $X_N^{(1)}$ и $X_N^{(2)}$. Конечно, точной разделимости не будет, но для практики это не так важно.

Как уже говорилось, недостатком описанного подхода является то, что метод не может быть применен к зашумленным рядам, поскольку он усиливает высокочастотные гармоники и, следовательно, усиливает шум. Поэтому в качестве предварительной обработки следует применять процедуру для удаления шума. Кроме того, SSA с использованием производных изменяет вклады компонентов (это то, что мы хотим), но одновременно метод теряет аппроксимационные свойства. Эти причины приводят к необходимости использования вложенного способа разложения, представленного в разделе 2.2.1.

Вложенный SSA с производными (DerivSSA)

Сформулируем вложенную версию SSA с производными, называемую DerivSSA. Как и в разделе 2.2.1, пусть $\mathbf{Y} = \mathbf{X}_I$ — одна из матриц в разложении $\mathbf{X} = \mathbf{X}_{I_1} + \dots + \mathbf{X}_{I_p}$, полученной на этапе группировки в Basic SSA; каждая группа соответствует отдельной компоненте временного ряда, и мы хотим построить уточненное разложение \mathbf{Y} . Как и раньше, обозначим $r = \text{rank } \mathbf{Y}$, $\mathbf{Y} = \mathcal{T}^{-1}\mathcal{H}\mathbf{Y}$.

АЛГОРИТМ 2.7: DerivSSA.

Input: Матрица \mathbf{Y} , вес производной $\gamma > 0$.

Output: разложение в уточненный ряд $\mathbf{Y} = \tilde{\mathbf{Y}}^{(1)} + \dots + \tilde{\mathbf{Y}}^{(l)}$.

- 1: Построить матрицу $\mathbf{Z} = [\mathbf{Y} : \gamma\Phi(\mathbf{Y})]$, где $\Phi(\mathbf{Y}) = [Y_2 - Y_1 : \dots : Y_K - Y_{K-1}]$.
- 2: Выполнить SVD для \mathbf{Z} : $\mathbf{Z} = \sum_{i=1}^r \sqrt{\lambda_i} U_i V_i^T$.
- 3: Построить разложение $\mathbf{Y} = \mathbf{X}_I$ в сумму элементарных матриц: $\mathbf{Y} = \sum_{i=1}^r U_i U_i^T \mathbf{Y}$.

-
- 4: Выбрать разбиение множества $\{1, \dots, r\} = \bigsqcup_{m=1}^l J_m$ и выполнить группировку для получения уточненного разложения матрицы $\mathbf{Y} = \mathbf{Y}_{J_1} + \dots + \mathbf{Y}_{J_l}$.
- 5: Получить уточненное разложение ряда $\mathbf{Y} = \tilde{\mathbf{Y}}^{(1)} + \dots + \tilde{\mathbf{Y}}^{(l)}$, где $\tilde{\mathbf{Y}}^{(m)} = \mathcal{T}^{-1}\mathcal{H}\mathbf{Y}_{J_m}$.
-

Заметим, что шаги 2 и 3 алгоритма корректны, так как пространство столбцов матрицы \mathbf{Z} совпадает с пространством столбцов матрицы \mathbf{Y} . Поэтому, $\text{rank } \mathbf{Z} = r$ и $\{U_i\}_{i=1}^r$ является ортонормированным базисом для пространства столбцов матрицы \mathbf{Y} .

Следующее предложение показывает, что алгоритм 2.7 является в точности алгоритмом 2.4 с конкретной парой матриц (\mathbf{L}, \mathbf{R}) , где $P_i = U_i$, Q_i — нормированные векторы $\mathbf{Y}^T U_i$ в (2.11).

Предложение 10. *Левые сингулярные векторы обычного SVD матрицы \mathbf{Z} совпадают с левыми сингулярными векторами (\mathbf{L}, \mathbf{R}) -SVD исходной матрицы \mathbf{Y} , где $\mathbf{L} \in \mathcal{M}_{L,L}$ — единичная матрица, а \mathbf{R} определяется равенством $\mathbf{R} = \mathbf{I}_K + \gamma^2 \mathbf{F}^T \mathbf{F}$, $\mathbf{I}_K \in \mathcal{M}_{K,K}$ — единичная матрица и*

$$\mathbf{F} = \begin{pmatrix} -1 & 1 & 0 & 0 & \cdots & 0 \\ 0 & -1 & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & -1 & 1 & 0 \\ 0 & \cdots & 0 & 0 & -1 & 1 \end{pmatrix} \in \mathcal{M}_{K-1,K}.$$

Доказательство. Заметим, что стандартное скалярное произведение в пространстве строк матрицы \mathbf{Z} может быть выражено как $(Z_1, Z_2)_{2K-1} = (Q_1, Q_2)_K + \gamma^2 (\Phi(Q_1), \Phi(Q_2))_{K-1}$, где Q_1 и Q_2 состоят из первых K элементов векторов Z_1 и Z_2 , $\Phi(Q) \in \mathbb{R}^{K-1}$ применительно к вектору $Q = (q_1, \dots, q_K)^T$ состоит из последовательных разностей компонент вектора $q_{i+1} - q_i$. Таким образом, если ввести скалярное произведение $\langle Q_1, Q_2 \rangle_{\mathbf{R}} = (\mathbf{R}Q_1, Q_2)_K$, то обычное SVD матрицы \mathbf{Z} может быть сведено к (\mathbf{L}, \mathbf{R}) -SVD матрицы \mathbf{Y} с соответствующими матрицами \mathbf{L} и \mathbf{R} . \square

Замечание 14. *Если \mathbf{Y} является траекторной матрицей ряда Y_N , то вложенный DerivSSA с производными эквивалентен реализации MSSA, описанной в разделе 2.3.1. Действительно, траекторная матрица производной временного ряда $\Phi_{N-1}(Y)$ совпадает с матрицей $\Phi(\mathbf{Y})$. Хотя, если \mathbf{Y} не является ганкелевой, то аналога метода в виде MSSA не существует.*

Разделимость

Пусть $\mathbf{X}_N = \mathbf{X}_N^{(1)} + \mathbf{X}_N^{(2)}$, где $\mathbf{X}_N^{(1)}$ и $\mathbf{X}_N^{(2)}$ конечного ранга и приближенно слабо разделимы, что подразумевает, что их строковые и столбцовые траекторные пространства приближенно ортогональны. То же самое справедливо и для $\Phi_{N-1}(\mathbf{X}^{(1)})$ и $\Phi_{N-1}(\mathbf{X}^{(2)})$, в силу того, что

их столбцовые траекторные пространства принадлежат столбцовыми траекторными пространствам рядов $X_N^{(1)}$ и $X_N^{(2)}$, а их строковые траекторные пространства являются линейными оболочками векторами той же структуры, что и векторы, составляющие базисы строковых траекторных пространств $X_N^{(1)}$ и $X_N^{(2)}$, за исключением того, что эти базисные векторы имеют длину $K - 1$, а не K . Поэтому приближенная ортогональность все еще имеет место. Так как $\Phi_{N-1}(X) = \Phi_{N-1}(X^{(1)}) + \Phi_{N-1}(X^{(2)})$, DerivSSA примененный к $(X_N, \gamma\Phi_{N-1}(X))$, будет приближенно разделять временной ряд на $X_N^{(1)}$ и $X_N^{(2)}$.

Таким образом, DerivSSA не ухудшает слабую разделимость и может достичь сильной разделимости. Важно всегда иметь в виду, что DerivSSA увеличивает вклад высокочастотных компонентов и уменьшает вклад низкочастотных компонентов.

2.3.2. Filter-adjusted O-SSA

Обратим внимание, что последовательные разности, которые берутся для каждой строки матрицы \mathbf{Y} , могут быть распространены на произвольный линейный фильтр строк. То есть, мы можем выбрать коэффициенты линейного фильтра $A = (a_1, \dots, a_t)^T$ и определить $\Phi(\mathbf{Y}) = [Y_1^*, \dots, Y_{K-t+1}^*]$, где $Y_i^* = a_1 Y_i + \dots + a_t Y_{i+t-1}$. Остальная часть соответствующей модификации SSA аналогична DerivSSA. DerivSSA соответствует $t = 2$ и $A = (-1, 1)^T$.

Алгоритм

Общий алгоритм Filter-adjusted O-SSA описан в двух эквивалентных формах в алгоритмах 2.8 и 2.9. Алгоритм 2.8 непосредственно реализует метод, приведенный в разделе 2.3.1, в то время как алгоритм 2.9 больше подходит для эффективной реализации и для модификации, реализованной в алгоритме 2.10.

АЛГОРИТМ 2.8: Filter-adjusted O-SSA: разложение

Input: Разложение L -траекторной матрицы $\mathbf{X} = \sum \sigma_i P_i Q_i^T$, группа компонент I , $|I| = r$, коэффициенты фильтра (a_1, \dots, a_t) , вес $\gamma > 0$.

Output: Разложение $\mathbf{Y} = \mathbf{X}_I$ по элементарным матрицам $\mathbf{Y} = \mathbf{Y}_1 + \dots + \mathbf{Y}_r$, где $\mathbf{Y}_i = \sigma'_i P'_i (Q'_i)^T$.

- 1: Сформировать матрицу $\mathbf{Y} = \mathbf{X}_I = \sum_{i \in I} \sigma_i P_i Q_i^T$.
- 2: Построить матрицу $\mathbf{Z} = [\mathbf{Y} : \gamma\Phi(\mathbf{Y})]$, где $\Phi(\mathbf{Y}) = [Y_1^*, \dots, Y_{K-t+1}^*]$ и $Y_i^* = a_1 Y_i + \dots + a_t Y_{i+t-1}$.
- 3: Вычислить SVD \mathbf{Z} : $\mathbf{Z} = \sum_{i=1}^r \sqrt{\lambda_i} U_i V_i^T$.
- 4: Построить следующее разложение $\mathbf{Y} = \mathbf{X}_I$ в сумму элементарных матриц: $\mathbf{Y} = \sum_{i=1}^r U_i U_i^T \mathbf{Y}$.
- 5: Получить разложение $\mathbf{Y} = \sum_{i=1}^r \sigma'_i P'_i (Q'_i)^T$, где $\sigma'_i = \|U_i^T \mathbf{Y}\|$, $P'_i = U_i$, $Q'_i = U_i^T \mathbf{Y} / \sigma'_i$.

АЛГОРИТМ 2.9: Filter-adjusted O-SSA: разложение (эквивалент)

Input: Разложение траекторной матрицы $\mathbf{X} = \sum \sigma_i P_i Q_i^T$, группа компонент I , $|I| = r$, коэффициенты фильтра (a_1, \dots, a_t) , вес γ .

Output: Разложение $\mathbf{Y} = \mathbf{X}_I$ по элементарным матрицам $\mathbf{Y} = \mathbf{Y}_1 + \dots + \mathbf{Y}_r$, где $\mathbf{Y}_i = \sigma'_i P'_i (Q'_i)^T$.

- 1: Сформировать матрицу $\mathbf{Y} = \mathbf{X}_I = \sum_{i \in I} \sigma_i P_i Q_i^T$ и вычислить ее неполное SVD $\mathbf{Y} = \mathbf{U}_r \Lambda_r^{1/2} \mathbf{V}_r^T$. Положить $\mathbf{S} = [S_1 : \dots : S_K] = \Lambda_r^{1/2} \mathbf{V}_r^T \in \mathbb{R}^{r \times K}$.
 - 2: Построить матрицу $\mathbf{Z} = [\mathbf{S} : \gamma \Phi(\mathbf{S})] \in \mathbb{R}^{r \times (2K-t+1)}$, где $\Phi(\mathbf{S}) = [S_1^*, \dots, S_{K-t+1}^*]$ и $S_i^* = a_1 S_i + \dots + a_t S_{i+t-1}$.
 - 3: Вычислить матрицу вращения $\tilde{\mathbf{U}} \in \mathbb{R}^{r \times r}$, состоящую из собственных векторов матрицы $\mathbf{Z} \mathbf{Z}^T$ в качестве столбцов.
 - 4: Положить $\tilde{\mathbf{P}} = [\tilde{P}_1 : \dots : \tilde{P}_r] = \mathbf{U}_r \tilde{\mathbf{U}}$ и $\tilde{\mathbf{Q}} = [\tilde{Q}_1 : \dots : \tilde{Q}_r] = \mathbf{S}^T \tilde{\mathbf{U}}$.
 - 5: Получить разложение $\mathbf{Y} = \sum_{i=1}^r \sigma'_i P'_i (Q'_i)^T$, где $\sigma'_i = \|\tilde{Q}_i\|$, $P'_i = \tilde{P}_i / \sigma'_i$, $Q'_i = \tilde{Q}_i / \sigma'_i$.
-

Метод, представленный в разделе 2.3.1, имеет модификацию, реализованную в RSSA, которая может немного ухудшить разделимость, но имеет то преимущество, что она упорядочивает собственные тройки, соответствующие синусоидам, именно по убыванию их частот, независимо от значений амплитуд синусоид. Мы будем называть эту модификацию ‘Filter-adjusted O-SSA with normalization’. Основное различие между алгоритмами 2.9 и 2.10 заключается в построении матрицы \mathbf{S} на шаге 1.

АЛГОРИТМ 2.10: Filter-adjusted O-SSA with normalization: разложение

Input: Разложение траекторной матрицы $\mathbf{X} = \sum \sigma_i P_i Q_i^T$, группа компонент I , $|I| = r$, коэффициенты фильтра (a_1, \dots, a_t) , вес γ .

Output: Разложение $\mathbf{Y} = \mathbf{X}_I$ по элементарным матрицам $\mathbf{Y} = \mathbf{Y}_1 + \dots + \mathbf{Y}_r$, где $\mathbf{Y}_i = \sigma'_i P'_i (Q'_i)^T$.

- 1: Сформировать матрицу $\mathbf{Y} = \mathbf{X}_I = \sum_{i \in I} \sigma_i P_i Q_i^T$ и построить ее косоугольное SVD $\mathbf{Y} = \mathbf{U}_r \Lambda_r^{1/2} \mathbf{V}_r^T$. Положить $\mathbf{S} = [S_1 : \dots : S_K] = \mathbf{V}_r^T \in \mathbb{R}^{r \times K}$.
 - 2: Построить матрицу $\mathbf{Z} = [\mathbf{S} : \gamma \Phi(\mathbf{S})] \in \mathbb{R}^{r \times (2K-t+1)}$, где $\Phi(\mathbf{S}) = [S_1^*, \dots, S_{K-t+1}^*]$ и $S_i^* = a_1 S_i + \dots + a_t S_{i+t-1}$.
 - 3: Вычислить матрицу вращения $\tilde{\mathbf{U}} \in \mathbb{R}^{r \times r}$, состоящую из собственных векторов $\mathbf{Z} \mathbf{Z}^T$ в качестве столбцов.
 - 4: Положить $\tilde{\mathbf{P}} = [\tilde{P}_1 : \dots : \tilde{P}_r] = (\mathbf{U}_r \Lambda_r^{1/2}) \tilde{\mathbf{U}}$ и $\tilde{\mathbf{Q}} = [\tilde{Q}_1 : \dots : \tilde{Q}_r] = \mathbf{S}^T \tilde{\mathbf{U}}$.
 - 5: Получить разложение $\mathbf{Y} = \sum_{i=1}^r \sigma'_i P'_i (Q'_i)^T$, где $\sigma'_i = \|\tilde{P}_i\|$, $P'_i = \tilde{P}_i / \sigma'_i$, $Q'_i = \tilde{Q}_i / \sigma'_i$.
-

Замечание 15. Алгоритмы 2.8–2.10 могут быть распространены на случай, когда приме-

няется несколько фильтров в сложенном виде. Например, если даны фильтры Φ_1 и Φ_2 , то матрица \mathbf{Z} на шаге 2 имеет вид $\mathbf{Z} = [\mathbf{Y} : \gamma\Phi_1(\mathbf{Y}) : \gamma\Phi_2(\mathbf{Y})]$ или $\mathbf{Z} = [\mathbf{S} : \gamma\Phi_1(\mathbf{S}) : \gamma\Phi_2(\mathbf{S})]$ соответственно.

Алгоритм восстановления такой же, как и для большинства версий SSA. Поскольку Filter-adjusted O-SSA (как и DerivSSA) является вложенным методом, результатом является разложение выбранной компоненты ряда, а не разложение исходного ряда.

АЛГОРИТМ 2.11: Filter-adjusted O-SSA: восстановление

Input: Разложение $\mathbf{Y} = \sum_{i=1}^r \mathbf{Y}_i$, где $\mathbf{Y}_i = \sigma'_i P'_i(Q'_i)^T$, группировка $I = \bigsqcup_{k=1}^l J_k$.

Output: Разложение определенного ряда $\mathbf{Y} = \tilde{\mathbf{Y}}^{(1)} + \dots + \tilde{\mathbf{Y}}^{(l)}$.

- 1: Получить сгруппированное разложение матрицы $\mathbf{Y} = \mathbf{Y}_{J_1} + \dots + \mathbf{Y}_{J_l}$, где $\mathbf{Y}_J = \sum_{j \in J} \mathbf{Y}_j$.
- 2: Получить уточненное разложение в ряд $\mathbf{Y} = \tilde{\mathbf{Y}}^{(1)} + \dots + \tilde{\mathbf{Y}}^{(l)}$, где $\tilde{\mathbf{Y}}^{(m)} = \mathcal{T}^{-1} \circ \Pi_{\mathcal{H}}(\mathbf{Y}_{J_m})$, $m = 1, \dots, l$.

Замечание 16. Алгоритм 2.11 описывает уточненное разложение матрицы \mathbf{X}_I . Однако, мы можем рассматривать Filter-adjusted O-SSA как алгоритм, который мы применяем к полному разложению траекторной матрицы \mathbf{X} исходного ряда X , который меняет компоненты из группы I . Тогда результат также является полным разложением \mathbf{X} .

2.3.3. Примеры

Разделение синусоид с равными амплитудами

Рассмотрим ряд $x_n = \sin(2\pi n/10) + \sin(2\pi n/15)$, $n = 1, \dots, N$, $N = 150$, $L = 70$, который изображен на рис. 2.10.

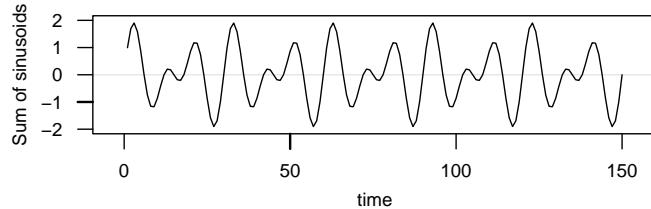


Рис. 2.10: Сумма двух синусоид с равными амплитудами.

Синусоиды с периодами 10 и 15 приближенно разделимы для таких рядов и длин окон. Однако, поскольку амплитуды синусоид равны, сильной разделимости нет, и поэтому после Basic SSA мы получаем неудовлетворительное разложение, произвольную смесь синусоид (верхний график на рис. 2.11) с w -корреляцией между восстановленными рядами ET1–2 и ET3–4 равной 0.92.

Разложение, выполненное DerivSSA с $\gamma = 10$, примененным к группе ET1–4 с $J_1 = \{1, 2\}$ и $J_2 = \{3, 4\}$ (алгоритм 2.7) изображено на нижнем графике рис. 2.11 и демонстрирует очень точную разделимость, w -корреляция равна 0.01. Второй показатель, средняя пропорция $0.5(\tau_{r_1}(X^{(1)}) + \tau_{r_2}(X^{(2)}))$, уменьшается с 0.3266 до 0.0003. Для данного примера полученное разложение практически не зависит от γ при всех $\gamma > 2$.

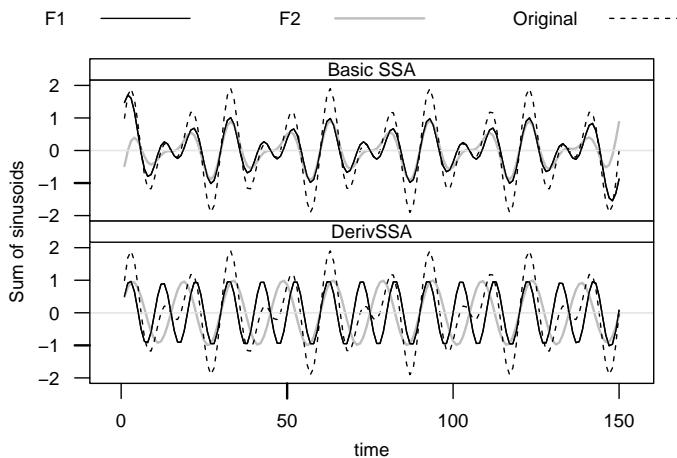


Рис. 2.11: Сумма двух синусоид с равными амплитудами: восстановление с помощью Basic SSA (вверху) и DerivSSA (внизу).

Реальные временные ряды

В этом разделе мы применяем Iterative O-SSA (алгоритмы 2.5 и 2.6) и DerivSSA (алгоритм 2.7) к реальным временным рядам. Роль методов для разделимости синусоид была продемонстрирована в разделах 2.2.4 и 2.3.3 с помощью сгенерированных данных. Полученные выводы в целом справедливы для реальных рядов: DerivSSA добавляет к Basic SSA способность разделять синусоиды с близкими амплитудами, в то время как Iterative O-SSA может помочь в разделении синусоид, которые не являются ортогональными, то есть их частоты недостаточно удалены одна от другой. Обратим внимание, что поскольку в реальных рядах с сезонностью не существует близких частот, DerivSSA может быть очень полезен для сезонного разложения.

В этом разделе мы рассмотрим проблему выделения тренда. Выбор примеров объясняется следующими соображениями.

Если временной ряд достаточно длинный, то колебания слабо отделены от тренда и только сильная разделимость находится под вопросом. Поэтому мы ожидаем, что DerivSSA будет работать для трендов сложной формы.

Для коротких рядов тренд может быть не ортогонален периодическому компоненту, например, сезонности; Поэтому DerivSSA может даже ухудшить разделимость; кроме того, дифференцирования подавляет низкочастотные компоненты. С другой стороны, Iterative O-SSA специально разработан для разделения неортогональных компонентов ряда.

В методе Iterative O-SSA мы сделаем только одну итерацию, так как этого достаточно для получения соответствующих разложений в рассматриваемых примерах, а также делает методы сравнимыми по вычислительным затратам.

Улучшение сильной разделимости Рассмотрим данные по безработице в США (ежемесячно, 1948–1981, в тысячах) для мужчин (20 лет и старше). Данные взяты из [119], длина ряда N равна 408, см. рис. 2.12. Поскольку ряд длинный, мы можем ожидать слабую разделимость тренда и сезонности. Для лучшей слабой разделимости мы выбираем длину окна равную $L = N/2 = 204$, которая кратна 12.

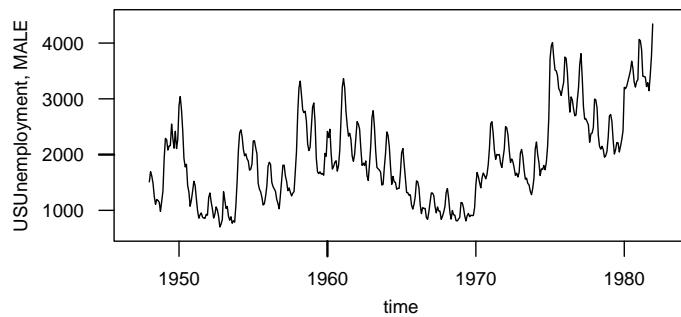


Рис. 2.12: Безработица в США: начальный ряд.

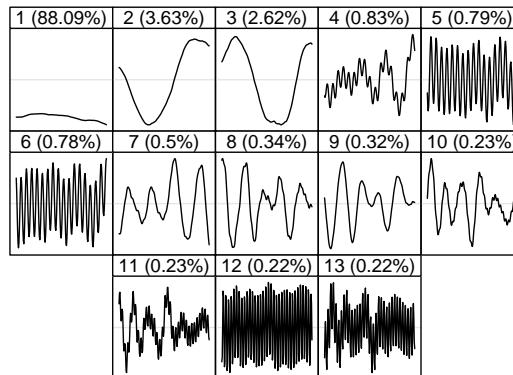


Рис. 2.13: Безработица в США: собственные векторы, полученные с помощью Basic SSA.

Basic SSA не разделяет тренд и сезонность для этого временного ряда (см. рис. 2.13 и рис. 2.17 (слева)), вероятно, из-за отсутствия сильной разделимости. Это типичная ситуация, когда тренд имеет сложную форму, компоненты тренда смешиваются с компонентами сезонности и поэтому ранее в литературе рекомендовался так называемый Sequential SSA [1, Раздел 1.7.3]. Однако это как раз тот случай, когда DerivSSA должен помочь.

Мы применяем DerivSSA к группе ET1–13, которая может быть связана с сигналом. DerivSSA разделяет различные частоты таким образом, чтобы что компоненты с более высокими частотами становятся ведущими. Поскольку низкочастотные компоненты в рассматриваемом ряду имеют большой вклад, вес производных должен быть большим, чтобы сезонные компоненты стали ведущими; мы возьмем $\gamma = 1000$.

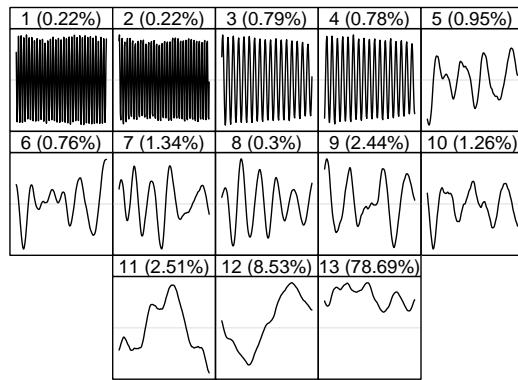


Рис. 2.14: Безработица в США: собственные векторы, полученные с помощью DerivSSA.

Полученные собственные векторы изображены на рис. 2.14. Видно, что первые 4 компоненты содержат сезонность, в то время как собственные векторы 5–13 содержат компоненты тренда. Смешение компонентов в группе тренда не имеет значения. Рис. 2.14 демонстрирует, что сезонные компоненты теперь отделены от остаточных. Рис. 2.15, изображающий DerivSSA восстановления тренда и сезонности, подтверждает, что DerivSSA заметно улучшает точность восстановления, особенно на концах ряда.

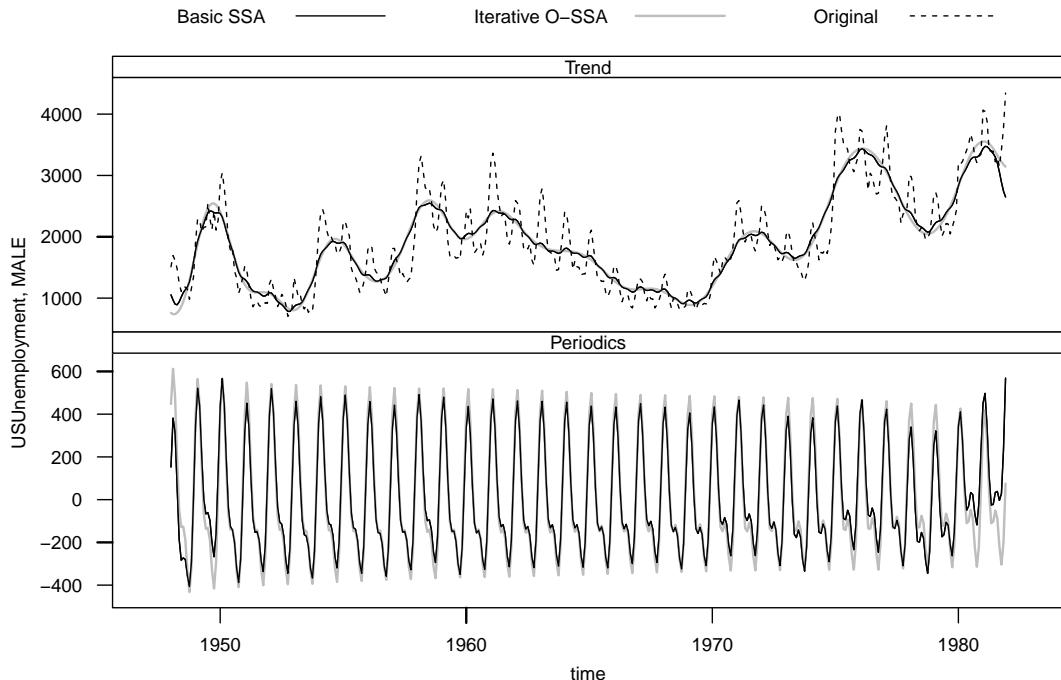


Рис. 2.15: Безработица в США: Разложения по Basic SSA и Iterative O-SSA, совпадающее с разложением по DerivSSA

Поскольку Iterative O-SSA может делать сигма-коррекцию, он также может раздвигать компоненты разложения, и поэтому мы можем применить Iterative O-SSA к группе ET1–13 с уточненными группами ET1–4,7–11 (тренд) и ET5,6,12,13 (сезонность). Поскольку компоненты разложения Basic SSA смешались, мы относим компоненты, содержащие в основном тренд

и медленные циклы, к первой группе, а компоненты, содержащие в основном сезонность, — ко второй. Поскольку собственные векторы отражают формы соответствующих компонентов временного ряда, мы можем использовать график собственных векторов, показанный на рис. 2.13, для первоначальной группировки. Например, четвертый собственный вектор выглядит как медленные колебания, испорченные сезонностью, и поэтому мы относим его к группе тренда, в то время как пятый собственный вектор выглядит как сезонный компонент, испорченный чем-то медленно меняющимся, и мы относим его к группе сезонности. Мы применяем одну итерацию с сигма-коррекцией, положив $\varkappa = 2$. После переупорядочения, вызванного сигма-коррекцией, первая группа тренда состоит из первых восьми компонентов 1–8, в то время как вторая группа сезонности состоит из 9–13 компонентов, см. рис. 2.16.

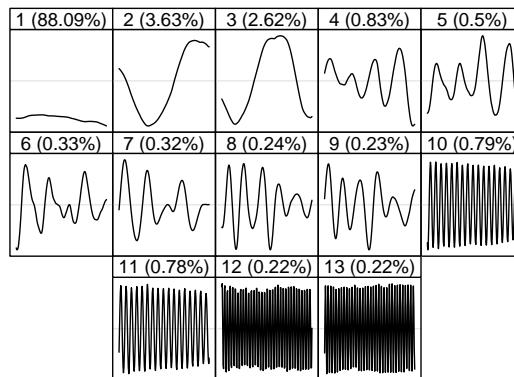


Рис. 2.16: Безработица в США: Собственные векторы Iterative O-SSA.



Рис. 2.17: Безработица в США: w -корреляции до (слева) и после (справа) Iterative O-SSA.

Трендовые собственные векторы разложения методом DerivSSA (рис. 2.14, ET5–13) отличаются от собственных векторов разложения методом Iterative O-SSA (рис. 2.16, ET1–8), компоненты сезонности почти одинаковы. Тем не менее, результат восстановления Iterative O-SSA визуально совпадает с результатом DerivSSA, показанным на рис. 2.15, и поэтому мы не изображаем это восстановление.

Рис. 2.17 содержит w -корреляции между элементарными восстановленными компонентами, предоставленными Basic SSA, (слева) и w -корреляции между элементарными восстановленными компонентами, полученными методом Iterative O-SSA (справа). Рисунок подтверж-



Рис. 2.18: Безработица в США: 2D графики периодических собственных векторов до (слева) и после (справа) Iterative O-SSA.

ждает улучшение разделимости. Обратим внимание, что хотя формально было получено ко-соугольное разложение, это разложение почти F-ортогонально (максимальная F-корреляция между элементарными компонентами матрицы, рассчитываемая, как $\langle \mathbf{X}_i, \mathbf{X}_j \rangle_F / (\|\mathbf{X}_i\|_F \|\mathbf{X}_j\|_F)$, равна 0.00368); поэтому подходят обычные \mathbf{w} -корреляции, см. раздел 2.2.1. Для выделения тренда важно, чтобы корреляции между трендом и сезонностью были близки к нулю. Действительно, корреляции между ET1–8 и ET9–13 малы. Смесь компонентов внутри группы трендов не важна. Видно, что компоненты тренда все еще слегка смешаны с компонентами шума. Это можно объяснить смешением с остатком до итераций (слева). Поэтому это не может быть исправлено Iterative O-SSA (справа), так как используется вложенная версия. Рис. 2.18 показывает улучшение разделимости с помощью диаграмм сезонных собственных векторов. После одной итерации графики сезонных собственных векторов образуют почти правильные многоугольники.

Рисунки для разложения DerivSSA, аналогичные рис. 2.17 и 2.18, очень похожи и мы не приводим их в данной работе. Заметим, что в DerivSSA мы группируем компоненты после их разделения, что проще, чем группировать смешанные компоненты для Iterative O-SSA до разделения. То есть, в рассмотренном примере результирующее разложение одинаково, но применение DerivSSA намного проще.

Улучшение слабой разделимости Рассмотрим ряд ‘Fortified wine’ (продажи крепленого вина, Австралия, ежемесячно, с января 1980 года, тысячи литров) взятый из [120]. Первые 120 точек ряда изображены на рис. 2.19.

Длина ряда достаточно велика, чтобы слабая разделимость имела место; поэтому мы будем рассматривать короткие подряды, чтобы продемонстрировать преимущество Iterative O-SSA для улучшения слабой разделимости.

Мы берем здесь длину окна $L = 18$, чтобы разница между методами была хорошо видна на рисунках, хотя соотношение между точностью рассматриваемых методов одинакова и при других вариантах длины окна. Рассмотрим два подряда, с 30-й по 78-ю точку и с 36-й по 84-ю точку. Разница заключается в поведении сезонности на концах подрядов.

Как и в предыдущем примере, мы начинаем с Basic SSA. Собственная тройка ET1 идентифицируется как соответствующая тренду, остальные компоненты порождены сезонностью

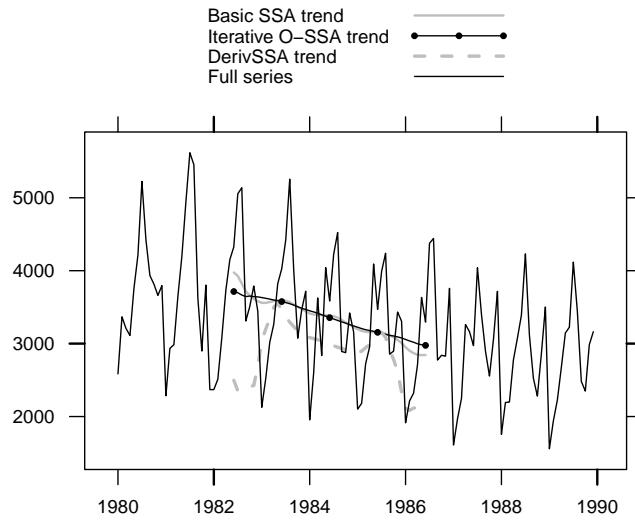


Рис. 2.19: Fortified wine: восстановление тренда с помощью DerivSSA и Iterative O-SSA для подрядов из точек 30–78.

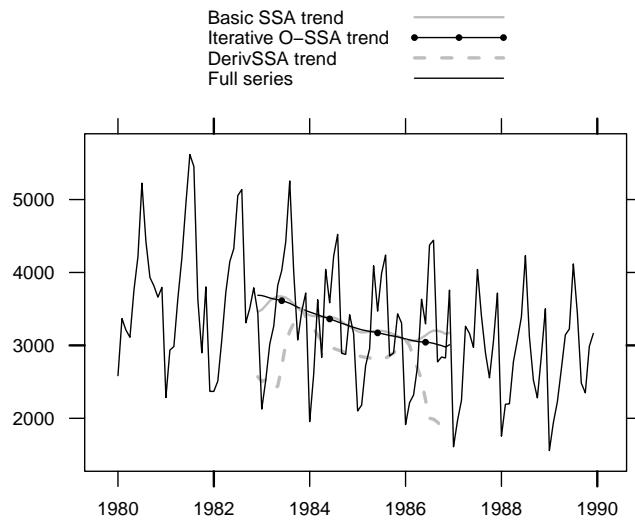


Рис. 2.20: Fortified wine: восстановление тренда с помощью DerivSSA и Iterative O-SSA для подрядов точек 36–84.

и шумом (мы не включаем их изображения). На рис. 2.19 и 2.20 (толстая серая линия) видно, что восстановленный тренд слегка смешивается с сезонностью и идет в сторону сезонной компоненты на концах ряда.

Чтобы применить Iterative O-SSA, мы должны выбрать группу элементарных компонентов, содержащих компоненты тренда и приближенно отделенных от остатка. Пусть это будет ET1–7. Таким образом, мы применяем одну итерацию O-SSA к уточненным группам ET1 и ET2–7. Поскольку вклад тренда намного больше, чем остатка, мы рассматриваем Iterative O-SSA без сигма-коррекции. Результат восстановления гораздо более релевантен, см. рис. 2.19 и 2.20 (сплошная линия с заполненными кружками). Серая пунктирная линия на тех же рисунках показывает, что DerivSSA дает восстановление хуже, чем Basic SSA в этом примере.

2.4. SSA–ICA

Этот раздел посвящен еще одному вложенному методу улучшения разделимости.

Анализ независимых компонент (Independent component analysis, ICA) [121, 122] — это статистический подход, который похож на PCA, но обеспечивает разложение на независимые случайные величины вместо некоррелированных компонентов. Целью ICA является нахождение статистически независимых компонентов $\{\eta_i; i = 1, \dots, p\}$ из наблюдаемых их линейных комбинаций $\{\xi_i; i = 1, \dots, p\}$. Конечно, для нормально распределенных случайных величин это одно и то же самое.

В этом разделе мы рассматриваем детерминированные задачи оптимизации, которые решаются в рамках выборочных версий метода ICA, чтобы построить разложение по матрицам единичного ранга; траекторная матрица рассматривается в качестве выборки.

Метод ICA не обладает свойствами аппроксимации, в частности, он не предназначен для отделения сигнала от шума. Поэтому метод SSA-ICA используется как вложенный метод, который уточняет разложение, полученное с помощью Basic SSA: сначала мы получаем оценку траекторной матрицы интересующего нас сигнала с помощью Basic SSA, а затем разлагаем полученную матрицу с помощью SSA-ICA.

Поскольку измерить статистическую зависимость сложно, существуют различные подходы к ICA. Мы рассмотрим два из них. Материалы разделов 2.4.1 и 2.4.2 содержатся в [29, раздел 1.5.4] и [14] соответственно.

2.4.1. Максимизация энтропии

Воспользуемся идеей метода поиска наилучшей проекции (projection pursuit) в многомерном анализе (обзор см. в [123]). Для выбора направлений поиск наилучшей проекции использует критерий, основанный на форме распределения проекции на заданное направление. Предполагая, что $L \leq K$, мы применим метод поиска наилучшей проекции к траекторной матрице, строки которой считаются признаками.

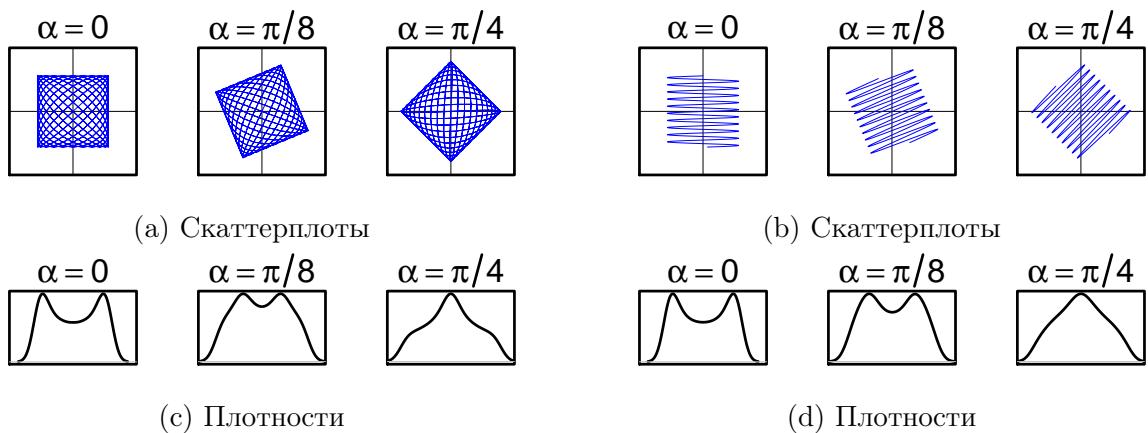


Рис. 2.21: Проекция: две синусоиды (слева) и синусоида с линейным рядом (справа).

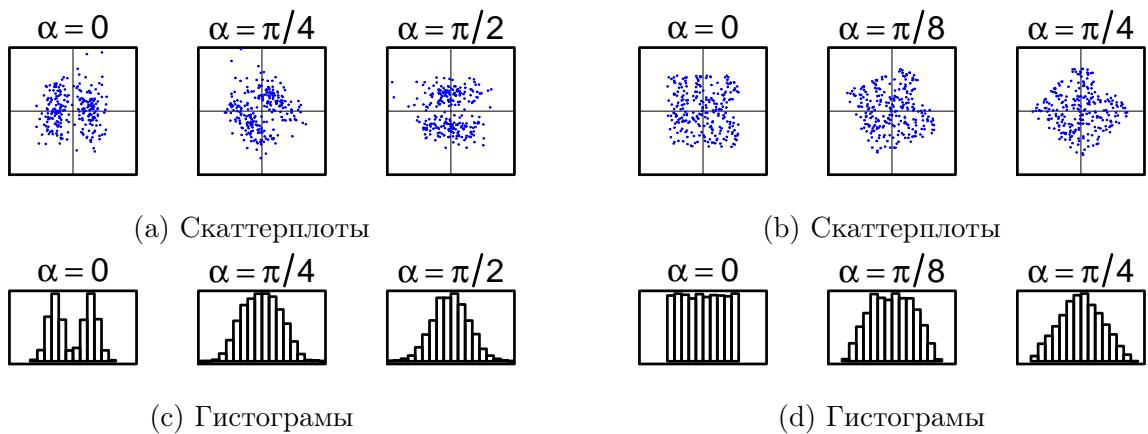


Рис. 2.22: Поиск наилучшей проекции: два кластера (слева) и две независимые равномерно распределенные переменные (справа).

Давайте начнем с рассмотрения проекций на различным направления для двух векторов, взятых из подпространств, соответствующих различным компонентам временного ряда. Для простоты изображения мы повернем данные и рассмотрим проекции на ось x. На рис. 2.21с показаны проекции для различных поворотов двух синусоидальных переменных. Первый рисунок в ряду (случай $\alpha = 0$) соответствует правильному вращению, последний (с $\alpha = \pi/4$) показывает наихудшую возможную смесь. Мы видим, что оцененные плотности совершенно разные. Чтобы проверить, что этот результат является общим, давайте рассмотрим аналогичные графики для синусоиды и линейной функции (рис. 2.21d). Результат очень похож. Таким образом, мы приходим к выводу, что идея метода поиска наилучшей проекции может помочь в решении проблемы разделения.

Рассмотрим метод поиска наилучшей проекции для кластерного анализа, где правильное вращение ($\alpha = 0$) соответствует максимальному расстоянию от нормального распределения. На рис. 2.22с показано, что распределения правильных проекций и неправильных проекций похожи на те, что изображены на рис 2.21с и 2.21d.

Одним из подходов к ICA является использование метода поиска наилучшей проекции (рис. 2.22d подтверждает это) [121, 122]. Опишем этот подход. Без потери общности мы можем предположить, что $\{\xi_i\}$ предварительно отбелены.

Взаимная информация случайного вектора (η_1, \dots, η_p) может быть измерена как

$$I(\eta_1, \dots, \eta_p) = \sum_{k=1}^p H(\eta_k) - H(\eta_1, \dots, \eta_p),$$

где $H(\eta) = - \int f(x) \log_2(f(x)) dx$ – дифференциальная энтропия и $f(x)$ – функция плотности η . Поэтому поиск независимых компонент эквивалентен поиску случайных величин $\{\eta_i\}$, которые являются линейными комбинациями $\{\xi_i\}$ и имеют минимальное значение взаимной информации.

Оказывается, что минимизация взаимной информации эквивалентна максимизации полноты информации.

ной негэнтропии $\{\eta_i\}$, которая является суммой маргинальных негэнтропий $J(\eta_i) = H(v) - H(\eta_i)$, $v \sim N(0, 1)$. Это означает, что ICA работает аналогично поиску направления с распределением проекции, максимально удаленным от нормального распределения, то есть, на поиск наилучших проекций.

Вместо того, чтобы максимизировать негэнтропию, что требует оценки маргинальных плотностей для вычисления энтропий η_i , мы можем рассмотреть максимизацию простых функционалов, таких как

$$J(\eta_i) \sim [-EG(\eta_i) + C_v]^2, \quad (2.12)$$

где $C_v = EG(v)$, $G(u) = e^{-u^2/2}$; можно рассмотреть и другие варианты G , см. [124]. Реализация ICA с помощью оптимизации функционала (2.12) представлена в пакете fastICA для R, см. [125].

Поскольку мы наблюдаем реализации p случайных величин $\mathbf{Y} = [Y_1 : \dots : Y_p]$, $Y_i \in \mathbb{R}^K$, вместо того, чтобы максимизировать (2.12) мы должны вычислить и максимизировать следующий функционал от их линейных комбинаций с коэффициентами $W \in \mathbb{R}^{p \times K}$:

$$J(Z) = \left(-\frac{1}{K} \sum_{i=1}^K e^{z_i^2/2} - C_v \right)^2 \rightarrow \max_{Z = \mathbf{Y}W, \|Z\|=1} \quad (2.13)$$

В приложениях к задаче слепого разделения сигналов взаимодействие между SSA и ICA уже рассматривалось, см. [126]. Там SSA используется для удаления шума, а затем ICA применяется для выделения независимых компонентов из смеси.

Теория метода ICA разрабатывалась для случайных величин и неприменима в детерминированном случае. Поэтому применение ICA к детерминированным данным можно формально рассматривать как поиск наилучших проекций, который ищет линейную комбинацию наблюдаемых переменных (факторных векторов в SSA), максимизируя некоторый функционал типа (2.13). Поскольку понятие статистической независимости не определено для детерминированных векторов мы будем использовать названия ‘ICA’ и ‘независимые векторы’ чисто формально и можем использовать кавычки при ссылках на них. Численным моделированием установлено и подтверждено теоретическими результатами, что в примерах, рассмотренных на рис 2.22 и подобных им, ‘ICA’ действительно удается разделить компоненты временного ряда, даже если SVD не обеспечивает сильной разделимости.

У ‘ICA’ есть следующий важный недостаток: он не делает упорядочения найденных компонентов (векторов), как это делает SVD. В частности, два вектора, соответствующие синусоиде, могут иметь произвольные номера в разложении методом ICA, и поэтому их поиск является более сложной задачей, чем при применении SVD. Кроме того, точность слабой разделимости, которую обеспечивает ICA, хуже, чем у SVD. Более того, стабильность численных процедур ICA хуже, чем у SVD. Поэтому рамках SSA метод ICA стоит рассматривать только в качестве дополнения к SVD для нахождения правильных вращений при наличии

слабой разделимости, но отсутствии сильной разделимости. Ни в коем случае ICA не может быть рекомендован в качестве полноценной замены SVD.

Ниже мы предлагаем схему построения уточненной группировки с помощью процедуры SSA-ICA. Эта схема может быть использована в качестве замены этапа группировки в Basic SSA. Пусть на этапе SVD мы имеем разложение $\mathbf{X} = \sum_{m=1}^d \sqrt{\lambda_j} U_j V_j^T$.

Уточненная группировка по SSA-ICA

1. Сделать группировку $\mathbf{X} = \mathbf{X}_{I_1} + \dots + \mathbf{X}_{I_m}$ как в Basic SSA, соответствующую сильно разделенным компонентам временного ряда.
2. Выбрать группу I , состоящую из p индексов, которая, возможно, состоит из нескольких интерпретируемых компонентов, которые перемешаны.
3. Извлечь p “независимых” векторов Q_i , применяя ‘ICA’ к \mathbf{X}_I . Тогда $\mathbf{X}_I = \sum_{i=1}^p P_i Q_i^T$, где $P_i = \mathbf{X}_I Q_i$.
4. Получить k подгрупп из группы I путем разбиения $\mathbf{X}_I = \mathbf{X}_{I,1} + \dots + \mathbf{X}_{I,k}$.

Пример 1. Рассмотрим пример ‘Fortified wines’, изображенный на рис. 2.19. Для анализа возьмем первые 120 точек. Длина окна L не обеспечивает сильную разделимость для ET8–11 (синусоиды с периодами 2.4 и 3), см. Рис. 2.23а, изображающий \mathbf{w} -корреляционную матрицу, где хорошо виден блок из четырех коррелированных компонент. Двухмерные диаграммы рассеяния факторных векторов изображены на рис. 2.23с и демонстрируют отсутствие структуры. Применим ‘ICA’ к траекторной матрице, восстановленной по собственным тройкам 8–11. На рис. 2.23б и 2.23д видно, что ‘ICA’ успешно разделяет две синусоиды. Отметим, что результирующие компоненты ‘ICA’ нуждаются в дополнительном упорядочивании, чтобы две синусоиды с одинаковой частотой получили последовательные индексы.

2.4.2. SOBI-AMUSE

Метод, рассмотренный в этом разделе, использует метод AMUSE, который является другой версией ICA, предложенной для разделения случайных сигналов [127, 128, 122], и больше подходит для анализа временных рядов.

Предположим, что мы наблюдаем ряд $\mathbf{X} = \mathbf{X}_N = (x_1, \dots, x_N)$ длины N , и $\mathbf{X}_N = \mathbf{X}_N^{(1)} + \mathbf{X}_N^{(2)}$.

Приведем алгоритм, который решает проблему отсутствия сильной разделимости слабо разделимых рядов. Слабо разделимые ряды остаются слабо разделимыми при слегка измененных условиях на длину ряда и длину окна.

В алгоритме SSA-AMUSE, помимо параметра длины окна L , существует параметр сдвига $\tau \geq 1$. Мы будем описывать алгоритм с произвольным τ . Однако, значения $\tau = 1$ достаточно для получения сильной разделимости.

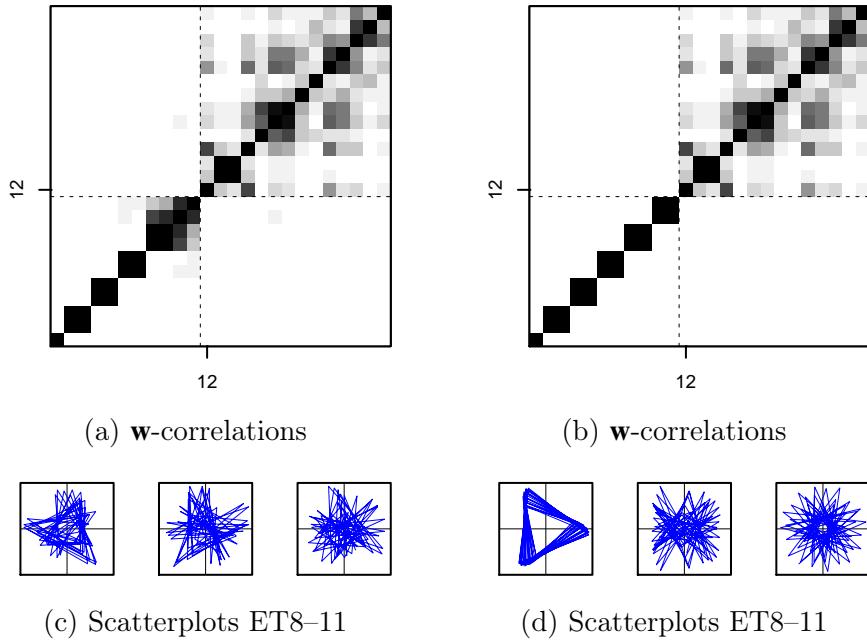


Рис. 2.23: ‘Fortified wines’: SVD (слева) и ICA для разделимости ET8-11 (справа).

Сначала приведем вспомогательные утверждения, необходимые для обоснования предлагаемого алгоритма. Введем обозначение, которое является ключевым для алгоритма и связывает его с методом AMUSE анализа независимых компонент.

Пусть \mathbf{B} — произвольная матрица из $\mathcal{M}_{A,B}$ и τ — некоторое целое положительное число. Обозначим через $\bar{\mathbf{B}}^{(\tau)}$ матрицу \mathbf{B} без первых τ строк, $\underline{\mathbf{B}}_{(\tau)}$ матрицу \mathbf{B} без последних τ строк, $(\tau) | \mathbf{B}$ матрица \mathbf{B} без первых τ столбцов, $\mathbf{B} |_{(\tau)}$ матрица \mathbf{B} без последних τ столбцов, когда $0 < \tau < A, 0 < \tau < B$.

Обозначим

$$\mathbf{C}_\tau(\mathbf{B}) = \frac{1}{2} \left[(\bar{\mathbf{B}}^{(\tau)})^T \underline{\mathbf{B}}_{(\tau)} + (\underline{\mathbf{B}}_{(\tau)})^T \bar{\mathbf{B}}^{(\tau)} \right]. \quad (2.14)$$

Вспомогательные утверждения

Лемма 5. Пусть $\mathbf{Y} \in \mathcal{M}_{L,K}$, $\tau < K - r$. Тогда для любого SVD матрицы $[(\tau) | \mathbf{Y} : \mathbf{Y} |_{(\tau)}] = \mathbf{U} \boldsymbol{\Lambda}^{1/2} \mathbf{T}^T$, где $\boldsymbol{\Lambda} \in \mathcal{M}_{r,r}$ — диагональная матрица с сингулярными числами на диагонали, $\mathbf{U} \in \mathcal{M}_{L,r}$ — матрица левых сингулярных векторов, и $\mathbf{T} \in \mathcal{M}_{2K-2\tau,r}$ — матрица правых сингулярных векторов, матрица \mathbf{T} имеет вид $\mathbf{T} = [\bar{\mathbf{Q}}^{(\tau)} : \underline{\mathbf{Q}}_{(\tau)}]$, где $\mathbf{Q} = \mathbf{Y}^T \mathbf{U} \boldsymbol{\Lambda}^{-1/2}$.

Доказательство. SVD матрицы $[(\tau) | \mathbf{Y} : \mathbf{Y} |_{(\tau)}]$ можно записать как

$$[(\tau) | \mathbf{Y} : \mathbf{Y} |_{(\tau)}] = \left[\mathbf{U} \boldsymbol{\Lambda}^{1/2} (\mathbf{T}^{(1)})^T : \mathbf{U} \boldsymbol{\Lambda}^{1/2} (\mathbf{T}^{(2)})^T \right],$$

где $\mathbf{T}^{(1)} = ((\tau) | \mathbf{Y})^T \mathbf{U} \boldsymbol{\Lambda}^{-1/2}$ и $\mathbf{T}^{(2)} = (\mathbf{Y} |_{(\tau)})^T \mathbf{U} \boldsymbol{\Lambda}^{-1/2}$. Тогда $\mathbf{Q} = \mathbf{Y}^T \mathbf{U} \boldsymbol{\Lambda}^{-1/2}$. Равенства $\bar{\mathbf{Q}}^{(\tau)} = \mathbf{T}^{(1)}$ и $\underline{\mathbf{Q}}_{(\tau)} = \mathbf{T}^{(2)}$ следуют из правил умножения матриц. \square

Замечание 17. Поскольку легко показать, что столбцы матрицы \mathbf{U} образуют ортонормированный базис пространства столбцов матрицы \mathbf{Y} , получаем $\mathbf{Y} = \mathbf{U}\mathbf{U}^T\mathbf{Y}$.

Следствие 4. $\mathbf{Y} = \mathbf{U}\Lambda^{1/2}\mathbf{Q}^T$.

Доказательство. Это следует из $\mathbf{Q} = \mathbf{Y}^T\mathbf{U}\Lambda^{-1/2}$ и замечания 17. \square

Заметим, что столбцы матрицы \mathbf{Q} не ортогональны и нормализованы.

Лемма 6. Пусть матрица $\mathbf{Q} \in \mathcal{M}_{K,r}$ и разложение $\mathbf{C}_\tau(\mathbf{Q}) = \mathbf{W}\mathbf{D}\mathbf{W}^T$ является спектральным. Тогда для $\mathbf{S} = \mathbf{Q}\mathbf{W}$ матрица $\mathbf{C}_\tau(\mathbf{S}) = \mathbf{D}$ диагональная.

Доказательство. Лемма получается путем подстановки $\mathbf{S} = \mathbf{Q}\mathbf{W}$ в $\mathbf{C}_\tau(\mathbf{S})$ и перемножения матриц. \square

Следующая теорема подтверждает разложение, полученное методом SSA-AMUSE.

Теорема 3. Пусть $\mathbf{Y} \in \mathcal{M}_{L,K}$ — матрица ранга r . Тогда $\mathbf{Y} = \sum_{i=1}^r \hat{U}_i S_i^T$, где \hat{U}_i — столбцы матрицы $\hat{\mathbf{U}} = \mathbf{U}\Lambda^{1/2}\mathbf{W}$, \mathbf{W} из спектрального разложения $\mathbf{C}_\tau(\mathbf{Q}) = \mathbf{W}\mathbf{D}\mathbf{W}^T$, $\mathbf{Q} = \mathbf{Y}^T\mathbf{U}\Lambda^{-1/2}$, \mathbf{U} — левые сингулярные векторы $[(\tau) | \mathbf{Y} | (\tau)]$.

Доказательство. По следствию 4 и лемме 6, $\mathbf{Y} = \mathbf{U}\Lambda^{1/2}\mathbf{W}\mathbf{S}^T = \hat{\mathbf{U}}\mathbf{S}^T = \sum_{i=1}^r \hat{U}_i S_i^T$, где \hat{U}_i — столбцы матрицы $\hat{\mathbf{U}} = \mathbf{U}\Lambda^{1/2}\mathbf{W}$. \square

Алгоритм

Как упоминалось выше, алгоритм SSA-AMUSE строит разложение некоторой сгруппированной матрицы $\mathbf{Y} = \mathbf{X}_I$, которая была получена на этапе разложения Basic SSA. Алгоритм построен следующим образом.

Алгоритм 2.12: SSA-AMUSE алгоритм

Input: Матрица \mathbf{Y} ранга r , сдвиг $\tau > 0$.

Output: Разложение $\mathbf{Y} = \mathbf{X}_I$ по элементарным матрицам $\mathbf{Y} = \mathbf{Y}_1 + \dots + \mathbf{Y}_r$, где $\mathbf{Y}_i = \hat{U}_i S_i^T$.

- 1: Построить сингулярное разложение матрицы $[(\tau) | \mathbf{Y} : \mathbf{Y} | (\tau)]$, которое по лемме 5 имеет вид $[(\tau) | \mathbf{Y} : \mathbf{Y} | (\tau)] = \mathbf{U}\Lambda^{1/2} \left[\left(\bar{\mathbf{Q}}^{(\tau)} \right)^T : \underline{\mathbf{Q}}^{(\tau)} \right]^T$, где $\mathbf{Q} = \mathbf{Y}^T\mathbf{U}\Lambda^{-1/2}$.
- 2: Получить $\mathbf{W} = [W_1 : \dots : W_r]$, где W_i , $i = 1, \dots, r$, являются собственными векторами матрицы $\mathbf{C}_\tau(\mathbf{Q})$.
- 3: Получить $\mathbf{S} = [S_1 : \dots : S_r] = \mathbf{Q}\mathbf{W}$.
- 4: Вычислить $\hat{\mathbf{U}} = [\hat{U}_1 : \dots : \hat{U}_r] = \mathbf{U}\Lambda^{1/2}\mathbf{W}$. По теореме 3, $\mathbf{Y} = \hat{\mathbf{U}}\mathbf{S}^T$. Таким образом, мы получаем разложение на элементарные матрицы $\mathbf{Y} = \sum_{i=1}^r \hat{U}_i S_i^T$.

Разделимость

Для удобства предположим, что траекторная матрица \mathbf{X} ряда X_N имеет ранг d . В качестве группы индексов I в алгоритмах мы будем брать номера ненулевых сингулярных чисел сингулярного разложения траекторной матрицы. Соответственно, сгруппированная матрица $\mathbf{Y} = \mathbf{X}_I = \mathbf{X}$ и имеет ранг $r = d$. В общем случае группа I может соответствовать произвольной компоненте ряда, сильно отделимой от остатка.

Вспомогательные утверждения Рассмотрим ряды конечного ранга, управляемые ЛРС, и докажем несколько утверждений о свойствах их траекторных пространств.

Обозначим $\mathfrak{X}^{(K,1)}$, $\mathfrak{X}^{(K,2)}$ и $\mathfrak{X}^{(K)}$ — пространства строк траекторных матриц временных рядов $X_N^{(1)}$, $X_N^{(2)}$ и X_N размерности d_1 , d_2 и d соответственно. Пусть $\mathfrak{X}^{(K,1)}|_{(\tau)}$, $\mathfrak{X}^{(K,2)}|_{(\tau)}$ и $\mathfrak{X}^{(K)}|_{(\tau)}$ — пространства строк матрицы без последних τ столбцов, а также $|_{(\tau)}|\mathfrak{X}^{(K,1)}$, $|_{(\tau)}|\mathfrak{X}^{(K,2)}$ и $|_{(\tau)}|\mathfrak{X}^{(K)}$ — пространства строк матрицы без первых τ столбцов.

Лемма 7. *Если ряд управляемся ЛРС ранга d , $d \leq \min(K, L - 2\tau)$, то $\mathfrak{X}^{(K)}|_{(\tau)} = |_{(\tau)}|\mathfrak{X}^{(K)}$ и $\dim(\mathfrak{X}^{(K)}|_{(\tau)}) = \dim(|_{(\tau)}|\mathfrak{X}^{(K)}) = \dim(\mathfrak{X}^{(K)}) = d$.*

Доказательство. Так как в матрице $\mathbf{X}|_{(\tau)}$ строки с $\tau + 1$ -й по L -ю совпадают со строками с первой по $L - \tau$ -ю матрицы $|_{(\tau)}|\mathbf{X}$, для доказательства равенства подпространств остается показать, что первые τ строки матрицы $\mathbf{X}|_{(\tau)}$ принадлежат пространству $|_{(\tau)}|\mathfrak{X}^{(K)}$, а строки с $(L - \tau + 1)$ -й по L -ю принадлежат $\mathfrak{X}^{(K)}|_{(\tau)}$. Действительно, это верно, так как строки подчиняется ЛРС, а указанные ряды являются линейными комбинациями нужных строк.

Теперь докажем равенство размерностей неполного и полного пространств. По доказанному выше $\dim(\mathfrak{X}^{(K)}|_{(\tau)}) = \dim(|_{(\tau)}|\mathfrak{X}^{(K)})$ и $\dim(\mathfrak{X}^{(K)}|_{(\tau)}) \leq \dim(\mathfrak{X}^{(K)})$. Рассмотрим базис V_1, \dots, V_d в $\mathfrak{X}^{(K)}$. Соответствующие векторы $\underline{V}_1|_{(\tau)}, \dots, \underline{V}_d|_{(\tau)} \in \mathfrak{X}^{(K)}|_{(\tau)}$ являются линейно независимыми, так как в противном случае они могли бы быть продолжены ЛРС до зависимых векторов, совпадающих по единственности с V_1, \dots, V_d . Лемма доказана. \square

Приведем еще одну лемму, которая будет использоваться при выводе сильных условий разделимости SSA-AMUSE.

Лемма 8. *Пусть матрица $\mathbf{S} = [S_1 : \dots : S_d]$ имеет ранг d , и для фиксированного $\tau > 0$ матрица $\mathbf{C}_\tau(\mathbf{S})$ диагональна: $\mathbf{C}_\tau(\mathbf{S}) = \text{diag}(\mu_1, \dots, \mu_d)$, $\mu_1 \geq \mu_2 \geq \dots \geq \mu_d$, матрица \mathbf{W} ортогональна, $\mathbf{T} = \mathbf{SW}^T$. Если для некоторых I_1 и I_2 множества $\{\mu_i, i \in I_1\}$ и $\{\mu_j, j \in I_2\}$ не пересекаются, то для любого спектрального разложения матрицы $\mathbf{C}_\tau(\mathbf{T}) = \mathbf{H}\mathbf{D}\mathbf{H}^T$ с собственными значениями на диагонали \mathbf{D} , упорядоченными по убыванию, справедливы следующие утверждения:*

1. $\mathbf{D} = \mathbf{C}_\tau(\mathbf{S})$,
2. $\text{span}(H_k : k \in I_1) = \text{span}(W_l : l \in I_1)$ и $\text{span}(H_k : k \in I_2) = \text{span}(W_l : l \in I_2)$, где $H_i, i = 1, \dots, d$, — столбцы матрицы \mathbf{H} , $W_j, j = 1, \dots, d$, — столбцы матрицы \mathbf{W} .

Доказательство. По лемме 6 матрицы $\mathbf{C}_\tau(\mathbf{S})$ и $\mathbf{C}_\tau(\mathbf{T})$ подобны, поэтому имеют одинаковые собственные значения, что доказывает первое утверждение леммы. Столбцы матрицы \mathbf{W} являются ортонормированными собственными векторами матрицы $\mathbf{C}_\tau(\mathbf{T})$. Поскольку подмножества собственных значений не пересекаются, собственные подпространства, соответствующие этим подмножествам, однозначно определены. Векторы W_i , $i \in I_1$, и W_j , $j \in I_2$, являются ортогональными базисами этих подпространств, а векторы H_i , $i \in I_1$, и H_j , $j \in I_2$, образуют другие ортогональные базисы тех же подпространств. \square

Условия разделимости Пусть $X_N = X_N^{(1)} + X_N^{(2)}$, $\mathbf{Y}^{(1)}$ и $\mathbf{Y}^{(2)}$ — траекторные матрицы временных рядов $X_N^{(1)}$ и $X_N^{(2)}$.

Легко видеть, что если пространства ${}_{(\tau)}|\mathfrak{X}^{(K,1)}$ и ${}_{(\tau)}|\mathfrak{X}^{(K,2)}$ строк матриц ${}_{(\tau)}|\mathbf{Y}^{(1)}$ и ${}_{(\tau)}|\mathbf{Y}^{(2)}$ ортогональны, то ряды $X_N^{(1)}$ и $X_N^{(2)}$ слабо SSA-AMUSE разделимы. Мы получим необходимые и достаточные условия для SSA-AMUSE разделимости.

Теорема 4. Ряды $X_N^{(1)}$ и $X_N^{(2)}$ слабо SSA-AMUSE разделимы тогда и только тогда, когда

$$\mathbf{Y}^{(1)}|_{(\tau)} \left[\mathbf{Y}|_{(\tau)}^{(2)} \right]^T + {}_{(\tau)}|\mathbf{Y}^{(1)} \left[{}_{(\tau)}|\mathbf{Y}^{(2)} \right]^T = 0, \quad (2.15)$$

$$\mathbf{Y}^{(1)}|_{(\tau)} \left[{}_{(\tau)}|\mathbf{Y}^{(2)} \right]^T + {}_{(\tau)}|\mathbf{Y}^{(1)} \left[\mathbf{Y}^{(2)}|_{(\tau)} \right]^T = 0. \quad (2.16)$$

Доказательство. Приведем схему доказательства.

Необходимость. Из отделимости следует, что существует базис $\{S_1, \dots, S_d\}$ пространства $\mathfrak{X}^{(K)}$, который распадается на подмножества $\{S_1^{(1)}, \dots, S_{d_1}^{(1)}\}$ и $\{S_1^{(2)}, \dots, S_{d_2}^{(2)}\}$ такие, что первое является базисом $\mathfrak{X}^{(K,1)}$, а второе — базисом $\mathfrak{X}^{(K,2)}$. Более того, если обозначить $\mathbf{S}^{(i)} = [S_1^{(i)} : \dots : S_{d_i}^{(i)}]$, $i = 1, 2$, $\mathbf{S} = [\mathbf{S}^{(1)} : \mathbf{S}^{(2)}]$ (для удобства мы переупорядочили векторы), тогда матрица $\mathbf{C}_\tau(\mathbf{S})$ диагональная. Обозначим $\Xi^{(i)} = \left[\left(\overline{\mathbf{S}^{(i)}}^{(\tau)} \right)^T : \left(\underline{\mathbf{S}^{(i)}}^{(\tau)} \right)^T \right]^T$. Столбцы матрицы $\Xi^{(1)}$ ортогональны столбцам $\Xi^{(2)}$, а столбцы матрицы $\Xi^{(i)}$ образуют ортогональный базис пространства строк матрицы ${}_{(\tau)}|\mathbf{Y}^{(i)}$. Отсюда следует равенство (2.15).

Используя условие разделимости, мы можем показать, что матрица

$$\mathbf{P}^{(i)} = \mathbf{U} \boldsymbol{\Lambda}^{1/2} \mathbf{W}^{(i)} \left(\mathbf{W}^{(i)} \right)^T \boldsymbol{\Lambda}^{-1/2} \mathbf{U}^T$$

определяет ортогональный проектор на пространство столбцов матрицы $\mathbf{Y}^{(i)}$. В свою очередь, используя это, мы выводим второе условие из диагональности матрицы $\mathbf{C}_\tau(\mathbf{S})$. Матрица может быть записана в виде

$$\mathbf{C}_\tau(\mathbf{S}) = \begin{pmatrix} \mathbf{C}_\tau(\mathbf{S}^{(1)}) & \mathbf{C}^{(1,2)} \\ \mathbf{C}^{(1,2)} & \mathbf{C}_\tau(\mathbf{S}^{(2)}) \end{pmatrix},$$

где

$$\mathbf{C}^{(1,2)} = \left(\mathbf{W}^{(1)} \right)^T \mathbf{\Lambda}^{-1/2} \mathbf{U}^T \left[\mathbf{Y}^{(1)}|_{(\tau)} \left(|_{(\tau)} \mathbf{Y}^{(2)} \right)^T + |_{(\tau)} \mathbf{Y}^{(1)} T \left(\mathbf{Y}^{(2)} \right) \right] \mathbf{U} \mathbf{\Lambda}^{-1/2} \mathbf{W}^{(2)} / 2$$

и равна нулю по условию диагональности. Умножим на $\left(\mathbf{W}^{(i)} \right)^T \mathbf{\Lambda}^{-1/2} \mathbf{U}^T$ справа и на $\mathbf{U} \mathbf{\Lambda}^{1/2} \mathbf{W}^{(i)}$ слева, затем используем свойства проектора $\mathbf{P}^{(i)}$ и получим условие (2.16).

Достаточность. По условию (2.15) и по лемме 5 существуют базисы $\mathbf{S}^{(1)} \in \mathfrak{X}^{(K,1)}$ и $\mathbf{S}^{(2)} \in \mathfrak{X}^{(K,2)}$ такие, что столбцы матрицы $\mathbf{\Xi}^{(1)}$ ортогональны друг другу и столбцам $\mathbf{\Xi}^{(2)}$. Отсюда можно сделать вывод, что столбцы матрицы $\mathbf{S} = [\mathbf{S}^{(1)} : \mathbf{S}^{(2)}]$ образуют базис $\mathfrak{X}^{(K)}$. Из условия (2.16) следует, что матрица $\mathbf{C}_\tau(\mathbf{S})$ является диагональной. Далее остается спроектировать столбцы траекторных матриц на найденный базис и найти элементарные матрицы: $\mathbf{Y}^{(i)} = \mathbf{Y}^{(i)} \mathbf{S}^{(i)} \left(\left(\mathbf{S}^{(i)} \right)^T \mathbf{S}^{(i)} \right)^{-1} \left(\mathbf{S}^{(i)} \right)^T$, что означает разделимость. \square

Следствие 5. *Если пространство $\mathfrak{X}^{(K,1)}|_{(\tau)}$ ортогонально $\mathfrak{X}^{(K,2)}|_{(\tau)}$, то ряды слабо SSA-AMUSE разделимы.*

Замечание 18. *По лемме 7, $\mathfrak{X}^{(K,i)}|_{(\tau)} =_{(\tau)} |\mathfrak{X}^{(K,i)}$. Поэтому вместо $\mathfrak{X}^{(K,i)}|_{(\tau)}$ в следствии 5, мы можем использовать $|_{(\tau)}|\mathfrak{X}^{(K,i)}$.*

Теорема 5. *Пусть временные ряды $X_N^{(1)}$ и $X_N^{(2)}$ слабо SSA-AMUSE разделимы. Обозначим $\mathbf{Y}^{(1)}$ и $\mathbf{Y}^{(2)}$ их траекторные матрицы. Пусть также $\mathbf{Q}^{(1)}$ — некоторый базис для первого ряда и $\mathbf{Q}^{(2)}$ — некоторый базис для второго ряда такой, что*

$$\left[|_{(\tau)} |\mathbf{Y}^{(i)} : \mathbf{Y}^{(i)}|_{(\tau)} \right] = \mathbf{U}^{(i)} \left(\mathbf{\Lambda}^{(i)} \right)^{1/2} \left[\left(\overline{(\mathbf{Q}^{(i)})}^{(\tau)} \right)^T : \left(\underline{(\mathbf{Q}^{(i)})}_{(\tau)} \right)^T \right], \quad i = 1, 2,$$

является сингулярным разложением. Тогда сильная SSA-AMUSE разделимость эквивалентна тому факту, что собственные числа матриц $\mathbf{C}_\tau(\mathbf{Q}^{(1)})$ и $\mathbf{C}_\tau(\mathbf{Q}^{(2)})$ не пересекаются.

Доказательство. Доказательство теоремы основано на лемме 8, согласно которой компоненты разложения, соответствующие разделяемым компонентам ряда, не могут быть смешаны. \square

Заметим, что условия сильной разделимости не зависят от выбора базисов $\mathbf{Q}^{(i)}$, $i = 1, 2$, удовлетворяющих условиям теоремы.

Следующее предложение показывает, как связаны условия слабой разделимости при применении Basic SSA и SSA-AMUSE. Его доказательство непосредственно следует из формулировки условий разделимости.

Предложение 11. *Пусть $X^{(1)}$ и $X^{(2)}$ — бесконечные ряды, управляемые ЛРС. Если отрезки длины N этих бесконечных рядов слабо разделимы в Basic SSA для длины окна L , то отрезки ряда, удлиненные на τ (то есть, длиной $N + \tau$) будут слабо SSA-AMUSE разделимы для тех*

же L и τ . Также, сегменты длины N будут слабо SSA-AMUSE разделимыми для длины окна $L - \tau$ и сдвига τ .

Замечание 19. Заметим, что для SSA-AMUSE разделимости необходима только ортогональность отрезков длиной $K - \tau$, тогда как для Basic SSA условия разделимости зависят от K и L .

Примеры слабой и сильной разделимости Приведем несколько примеров, показывающих особенности и преимущества разделимости методом SSA-AMUSE. Ниже мы рассмотрим только случай $\tau = 1$, так как этого достаточно, чтобы показать преимущества метода SSA-AMUSE.

1. Вот пример того, что условие ${}_{(\tau)}|\mathfrak{X}^{(K,1)} \perp {}_{(\tau)}|\mathfrak{X}^{(K,2)}$ является только достаточным условием для слабой разделимости. Рассмотрим $X^{(1)} = (\cos \pi n, n = 1, \dots, N)$ и $X^{(2)} = (\text{const}, n = 1, \dots, N)$. Если K нечетно, то пространства не ортогональны, но условия (2.15) и (2.16) выполняются. Таким образом, эти ряды слабо SSA-AMUSE разделимы.
2. Получим условия для сильной разделимости двух гармоник. Рассмотрим ряды $X^{(1)}$ и $X^{(2)}$ длины N с общими членами в виде $x_n^{(i)} = A_i \sin(2\pi n \omega_i + \gamma_i)$, $n = 1, \dots, N$, $0 < \omega_i \leq 0.5$, $\omega_1 \neq \omega_2$. Докажем, что если $(K - 1)\omega_1$ и $(K - 1)\omega_2$ — целые числа, то ряды сильно SSA-AMUSE разделимы. Слабая SSA-AMUSE разделимость следует из слабой разделимости в Basic SSA (см. утверждение 11). Сильная разделимость может быть доказана следующим образом. Пусть $\omega_i < 0.5$ для определенности. В этом случае базисы, найденные на шаге 3 алгоритма SSA-AMUSE, будут ортогональными и иметь вид:

$$\begin{aligned} S_1^{(i)} &= (A \sin(2\pi \omega_i + \zeta_i), \dots, A \sin(2\pi \omega_i K + \zeta_i))^T, \\ S_2^{(i)} &= (A \cos(2\pi \omega_i + \zeta_i), \dots, A \cos(2\pi \omega_i K + \zeta_i))^T, \end{aligned}$$

где $A = 1/\sqrt{K - 1}$.

Более того, для таких базисов каждая из матриц $\mathbf{C}_\tau(S^{(i)})$, $i = 1, 2$, является диагональной с числами $\cos(2\pi \omega_i)/2$ на диагонали, которые также являются собственными числами этих матриц. Поэтому условия сильной разделимости в теореме 5 удовлетворяются.

Заметим, что в случае SSA-AMUSE компоненты будут упорядочены по величинам $\cos(2\pi \omega_i)$, тогда как в Basic SSA они упорядочены по амплитудам A_i , а именно упорядочение по A_i приводит к смешиванию в случае совпадения амплитуд.

Асимптотическая SSA-AMUSE разделимость В реальных задачах редко выполняются точные условия разделимости, поэтому имеет место приближенная разделимость, что является следствием асимптотической разделимости.

Понятие асимптотической слабой SSA-AMUSE разделимости определяется аналогично асимптотической разделимости в случае Basic SSA путем замены точного условия ортогональности на асимптотическое, когда длина ряда N стремится к бесконечности. В случае SSA-AMUSE не требуется, чтобы L стремилось к бесконечности, и достаточно, чтобы только $K = N - L + 1 \rightarrow \infty$ при $N \rightarrow \infty$ (см. примечание 19).

Таким образом, те ряды, которые были слабо асимптотически разделимы методом Basic SSA остаются слабо асимптотически разделимыми методом SSA-AMUSE, при этом необязательно, чтобы L стремилось к бесконечности.

Рассмотрим понятие асимптотической сильной разделимости. Условием сильной разделимости в SSA-AMUSE является дизъюнкция множеств собственных значений матриц $\mathbf{C}_\tau(\mathbf{S}^{(K,i)})$, где $\mathbf{S}^{(K,i)}$ — матрицы, столбцы которых содержат базисы, найденные на шаге 3 алгоритма SSA-AMUSE. Условием асимптотической сильной разделимости асимптотически слабо разделимых рядов назовем дизъюнкцию множества предельных собственных значений матриц $\mathbf{C}_\tau(\mathbf{S}^{(K,i)})$ при $K \rightarrow \infty$, если такие пределы существуют.

Гармоники $X^{(1)}$ и $X^{(2)}$ с общими членами в виде $x_n^{(i)} = A_i \sin(2\pi n \omega_i + \gamma_i)$, $n = 1, \dots, N$, асимптотически сильно разделимы даже с совпадающими амплитудами при $K \rightarrow \infty$. Это доказывается прямым вычислением собственных значений матриц $\mathbf{C}_\tau(\mathbf{S}^{(K,i)})$.

Сравнение

Давайте сравним метод SSA-AMUSE и метод DerivSSA, описанный в разделе 2.3.1. У этих методов много общего. В частности, они оба являются вложенными, т.е. используются уже после выделения сигнала из шума, а также оба служат для улучшения сильной разделимости.

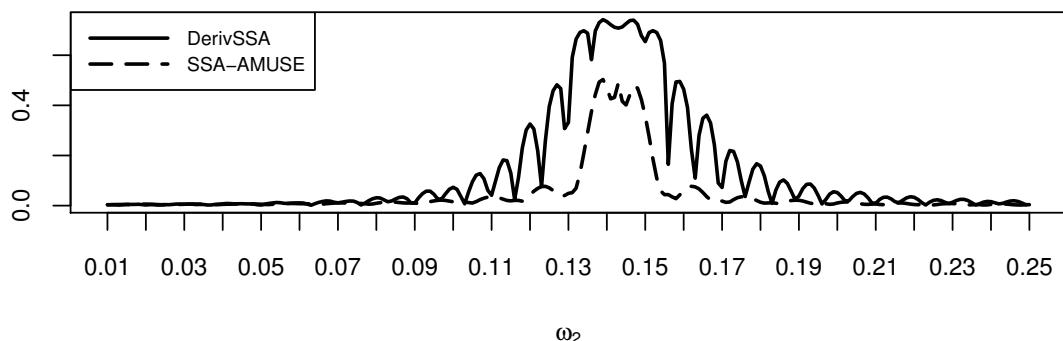


Рис. 2.24: RMSE оценок ряда с общим членом $\sin(2\pi n/7)$ по наблюдаемой сумме $\sin(2\pi n/7) + \sin(2\pi n\omega_2)$ в зависимости от ω_2 .

Одним из преимуществ SSA-AMUSE перед DerivSSA является возможность точного

разделения компонентов, что недоступно в DerivSSA. Некоторым недостатком SSA-AMUSE является то, что полученное разложение не является разложением на ортогональные матрицы Фробениуса.

Тем не менее, перечисленные выше преимущества приводят к повышению точности разделения компонентов временного ряда. Продемонстрируем это на численном эксперименте. На рис. 2.24 показаны ошибки RMSE оценки одной гармонической составляющей по наблюдаемой сумме двух синусов. Параметр ω_2 изменяется от 0.01 до 0.25 с шагом 0.001. Естественно, разделимость отсутствует, когда $\omega_2 \approx \omega_1 = 1/7$. Однако видно, что область с большой ошибкой около $\omega_2 = 1/7 \approx 0.14$ для SSA-AMUSE более узкая. Когда к сигналу добавляется шум, результат сравнения аналогичен.

2.5. Автоматическая идентификация

В этом разделе мы рассмотрим проблему автоматической идентификации компонентов разложения в SSA, см. недавнюю работу [129]. Начнем с определения компонентов временного ряда, которые мы хотим идентифицировать.

Рассмотрим временной ряд $X = (x_1, \dots, x_N)$, $x_i \in \mathbb{R}$.

Мы будем называть *тренд* низкочастотной компонентой. Для одномерного вещественного временного ряда это компонента T , для которой в разложении Фурье ряда T

$$t_n = C_0 + \sum_{k=1}^{\lfloor (N-1)/2 \rfloor} \sqrt{C_k^2 + S_k^2} \cos(2\pi nk/N + \phi_k) + C_{N/2}(-1)^n$$

наибольшие значения имеют коэффициенты $\sqrt{C_k^2 + S_k^2}$ при малом значении k ; последнее слагаемое $C_{N/2}(-1)^n$ присутствует только при четном N .

Колебательные компоненты будем называть различными регулярными колебаниями. Формально мы будем рассматривать сумму экспоненциально-модулированных (э.-м.) гармонических рядов (гармоник). Для одномерного вещественного ряда n -й элемент э.-м. гармоники с частотой ω ($\omega \leq 0,5$) дается выражением: $a e^{\alpha n} \cos(2\pi \omega n + \phi)$, $0 \leq \phi < 2\pi$, $a \neq 0$.

Сигналом будем называть сумму тренда и колебательной компоненты.

Мы предполагаем, что временной ряд X содержит следующие аддитивные компоненты: тренд T , осциллирующую компоненту P и случайный шум N . Таким образом, в общем случае рассматриваемая модель объекта X имеет вид:

$$X = T + P + N,$$

где элементы N являются реализациями случайной величины.

2.5.1. Метод низких частот для идентификации тренда

Рассмотренный метод автоматизации был представлен в [42]. Подход к автоматическому выбору параметров метода автоматизации предложен в [130], здесь мы его рассматривать не будем. Метод автоматизации позволяет выявить компоненты, связанные с трендом. Мы будем называть этот метод *низкочастотным методом для идентификации тренда*.

Введем в рассмотрение периодограмму:

$$\Pi_{\mathbf{Y}}^M(k/M) = \frac{M}{2} \begin{cases} 2C_0^2 & \text{для } k = 0, \\ C_k^2 + S_k^2 & \text{для } 0 \leq k \leq M/2, \\ 2C_{M/2}^2 & \text{для } k = M/2, \text{ если } M \text{ четное}, \end{cases} \quad (2.17)$$

где коэффициенты C_k и S_k взяты из разложения Фурье ряда $\mathbf{Y} = (y_1, \dots, y_M)$:

$$y_n = C_0 + \sum_{k=1}^{\lfloor M/2 \rfloor} \left(C_k \cos(2\pi n k/M) + S_k \sin(2\pi n k/M) \right).$$

Для ряда \mathbf{Y} длины M и для $0 \leq \omega_1 \leq \omega_2 \leq 0.5$, определим

$$T(\mathbf{Y}; \omega_1, \omega_2) = \sum_{k: \omega_1 \leq k/M < \omega_2} I_{\mathbf{Y}}^M(k/M), \quad (2.18)$$

где

$$I_{\mathbf{Y}}^M(k/M) = \Pi_{\mathbf{Y}}^M(k/M) / \|\mathbf{Y}\|^2, \quad (2.19)$$

$\Pi_{\mathbf{Y}}^M$ определена в (2.17). Так как у нас $\|\mathbf{Y}\|^2 = \sum_{k=1}^{\lfloor M/2 \rfloor} \Pi_{\mathbf{Y}}^M(k/M)$, меру $T(\mathbf{Y}; \omega_1, \omega_2)$ можно рассматривать как долю частот, содержащихся в частотном диапазоне $[\omega_1, \omega_2]$.

Одной из целей при выполнении группировки является выделение компонента ряда с частотным диапазоном в основном из выбранного частотного диапазона. Поэтому естественно вычислить значение T для элементарных восстановленных компонентов. Более того, восстановление методом SSA можно рассматривать как линейный фильтр. Оказывается, что частотная характеристика фильтра, генерируемого i -ой собственной тройкой, почти совпадает с периодограммой соответствующего сингулярного вектора, см. [29, Предложение 3.13]. Поэтому целесообразно применять T и к сингулярным векторам для восстановления компонентов ряда с заданными частотными диапазонами.

Поскольку тренд ряда можно определить как медленно меняющуюся компоненту ряда, то для выделения тренда следует выбрать частотный диапазон в виде $[0, \omega)$. Поэтому будем рассматривать

$$T(\mathbf{Y}; \omega) = \sum_{k: k/M < \omega} I_{\mathbf{Y}}^M(k/M). \quad (2.20)$$

Значение ω отражает диапазон частот, который мы ассоциируем с трендом. Например, если ряд имеет ежемесячную сезонность, ω должна быть заметно меньше, чем $1/12$. Обратим внимание, что метод группировки не дает ответа на вопрос, является ли выделенный компонент действительно детерминированным трендом или просто результатом сглаживания.

Значения T для каждого элементарного компонента разложения могут быть использованы для выполнения группировки. Для автоматической группировки необходимо задать порог T_0 , $0 \leq T_0 \leq 1$. Например, если значение $T(Y_i; \omega)$ больше, чем T_0 для некоторого малого ω , где Y_i — i -й элементарный ряд или i -й левый/правый сингулярный вектор, то соответствующая собственная тройка может автоматически рассматриваться как часть тренда.

Предлагаемый метод представлен в алгоритме 2.13.

АЛГОРИТМ 2.13: 1D-SSA: Частотные диапазоны, идентификация по порогу

Input: Диапазон частот $[\omega_1, \omega_2]$, порог T_0 , группа I , тип ряда: собственные векторы, факторные векторы или восстановленный ряд.

Output: Группа компонентов $J \subset I$.

- 1: Для каждого ряда Y_i , $i \in I$, вычислить меру $T(Y_i; \omega_1, \omega_2)$, данная в (2.18).
- 2: Получить результирующую группу J из индексов $i \in I$ таких, что $T(Y_i; \omega_1, \omega_2) \geq T_0$.

2.5.2. Частотный метод идентификации колебательной компоненты

Будем определять колебательную компоненту как сумму э.-м. гармоник. Рассмотрим алгоритм автоматической идентификации э.-м. гармоник. Мы будем называть этот метод *частотный метод идентификации колебательной составляющей*.

Метод был предложен в [64] и развит в [36] и [130]; он основан на исследовании периодограмм сингулярных векторов, соответствующих ряду. Метод состоит из двух частей, на первом этапе проводится предварительная проверка, а выявленные на этом этапе сингулярные тройки проверяются на втором этапе.

Как известно из предложения 1, э.-м. гармоника (1.18) может иметь ранг 1, если частота $\omega = 0.5$, или ранг 2 в противном случае, т.е. э.-м. гармонике может соответствовать либо один сингулярный вектор, либо два.

Рассмотренный метод автоматической идентификации компонент, соответствующих э.-м. гармоникам, проверяет

- каждую пару сингулярных векторов, если они похожи на э.-м. гармоники с одинаковой частотой,
- каждый сингулярный вектор, если он похож на э.-м. гармонику с периодом 2 ($\omega = 0.5$).

Замечание 20. Поскольку при достаточно общих условиях одномерный вещественный гармонический ряд с $\omega < 0.5$ порождает два одинаковых (по предложению 2) или близких

([1]) собственных числа траекторной матрицы, а алгоритм метода SSA сортирует компоненты сингулярного разложения по значению собственных чисел, для идентификации гармоник ранга 2 достаточно рассматривать только последовательные пары сингулярных векторов.

Согласно разделу 1.8.4, гармоническому ряду вида (1.18) соответствуют два сингулярных вектора вида (1.19) с одинаковыми частотами.

Известен следующий результат, где нормированная периодограмма I_Y^M задана в (2.19).

Предложение 12. [130, Предложение 3.1] Пусть $\alpha = 0$, $\omega < 0.5$ и $L\omega \in \mathbb{N}$. Тогда для сингулярных векторов U_1 , U_2 временного ряда S с элементами, заданными (1.18)

$$\max_{0 \leq k \leq L} I_{U_1}^L(k/L) = \max_{0 \leq k \leq L} I_{U_2}^L(k/L) = I_{U_1}^L(\omega) = I_{U_2}^L(\omega) = 1.$$

Первый шаг метода основан на том, что если для гармоники ($\alpha = 0$) с частотой ω и длиной окна L выполняется соотношение $L\omega \in \mathbb{N}$, то периодограммы его сингулярных векторов $I_{U_j}^L(k/L) = \chi_\omega(k/L)$ (где $\chi_\omega(\cdot)$ — индикатор множества $\{\omega\}$). Учитывая лишь приближенную разделимость когда $L\omega \notin \mathbb{N}$ или $\alpha \neq 0$, на первом этапе метода из всех последовательных пар сингулярных векторов выбираются те, для которых аргументы максимумов периодограммы больше 0 и близки, т.е. отличаются друг от друга не более чем на s_0/L (где $s_0 \in \mathbb{Z}_+$ — фиксированный параметр метода):

$$J_1^{(P)} = \{(i, i+1) : \theta_1, \theta_2 > 0, L|\theta_i - \theta_{i+1}| \leq s_0, 1 \leq i \leq d-1\}, \quad (2.21)$$

где $\theta_j = \arg \max_{0 < k \leq L/2} \{I_{U_j}^L(k/L)\}$ — аргумент максимума периодограммы $I_{U_j}^L$ сингулярного вектора U_j .

Аналогично, каждый сингулярный вектор проверяется на гармонику с частотой 0.5:

$$J_2^{(P)} = \{i : L|\theta_i - 0.5| \leq s_0, 1 \leq i \leq d\}. \quad (2.22)$$

Результатом первого шага являются множества $J_1^{(P)}$ и $J_2^{(P)}$ номеров сингулярных векторов. Множество $J_1^{(P)}$ состоит из пар номеров сингулярных векторов, идентифицированных как соответствующие э.-м. гармоникам с $0 < \omega < 0.5$. Множество $J_2^{(P)}$ содержит числа сингулярных векторов, идентифицированных как соответствующие э.-м. гармоникам с $\omega = 0.5$.

Определение 9. Пусть $\mathcal{A} = \{W_j\}$, W_j — конечное множество вещественных векторов, обозначим мощность множества как $\#\mathcal{A}$. Введем функцию $\rho_{\mathcal{A}}$, определенную на множестве \mathcal{A} :

$$\rho_{\mathcal{A}}(k/L) = \frac{1}{\#\mathcal{A}} \sum_{W_j \in \mathcal{A}} I_{W_j}^L(k/L)$$

Ввиду предложения 12, чтобы проверить, что сингулярные векторы с номерами $j, j+1$ соответствуют гармонике, мы должны убедиться, что максимумы их периодограмм достигаются в одной точке, и что каждая из них имеет значение, близкое к 1, так как э.-м. гармонике с частотой ω такой, что $L\omega \notin \mathbb{N}$, также соответствует большое значение максимума периодограммы (хотя и меньшее 1), потому что ее периодограмма имеет один отчетливый пик. Это и проверяется на втором этапе метода.

До статьи [36], на основе которой мы описываем метод, рассматриваемый подход был предложен в [64], но сама идея не нова и была предложена еще в 1929 году Фишером в [131].

Таким образом, второй шаг метода заключается в выборе порога ρ_0 для меры

$$\rho_{i,j} := \max_{0 < k \leq L/2} \left(\rho_{\{U_i, U_j\}}(k/L) + \rho_{\{U_i, U_j\}}((k+1)/L) \right), \quad (2.23)$$

где U_i, U_j — пара сингулярных векторов с индексами из набора индексов $J_1^{(\mathcal{P})}$, выбранных на первом этапе.

Чтобы определить компоненты, принадлежащие гармонике с частотой 0.5, введем меру в виде

$$\rho_i := \rho_{\{U_i\}}((\lfloor L/2 \rfloor)/L) + \rho_{\{U_i\}}((\lfloor L/2 \rfloor + 1)/L), \quad (2.24)$$

где $i \in J_2^{(\mathcal{P})}$.

Конечным результатом частотного метода, реализуемого алгоритмом 2.14, являются индексы

$$J^{(\mathcal{P})} = \{(i, j) \in J_1^{(\mathcal{P})} : \rho_{i,j} \geq \rho_0\} \cup \{i \in J_2^{(\mathcal{P})} : \rho_i \geq \rho_0\}. \quad (2.25)$$

АЛГОРИТМ 2.14: 1D-SSA. Частотный метод идентификации колебательной компоненты

Input: Данные и параметры следующие.

1. **Данные:** левые сингулярные векторы $\{U_i\}_{i=1}^d$.
2. **Параметры:** параметр $s_0 \in \mathbb{Z}_+$, порог $\rho_0 \in [0, 1]$.

Output: Группа индексов компонент $J^{(\mathcal{P})}$, связанных с колебательной компонентой.

- 1: На основе $\{U_i\}_{i=1}^d$ получить индексную группу $J_1^{(\mathcal{P})}$, используя (2.21) с s_0 , и индексную группу $J_2^{(\mathcal{P})}$, используя (2.22) с s_0 .
- 2: Получить индексную группу $J^{(\mathcal{P})}$ с помощью (2.25), примененного к $\{U_j\}_{j \in J_1^{(\mathcal{P})} \cup J_2^{(\mathcal{P})}}$ с ρ_0 .

Автор [130] советует брать $s_0 = 1$. Недостатком метода является тот факт, что он изначально разработан для определения немодулированных гармоник. При применении к моду-

лированным гармоникам порог должен быть уменьшен [36], что, правда, может привести к случаям ложного обнаружения гармоник в немодулированном случае ($\alpha = 0$).

2.5.3. Метод идентификации колебательной компоненты по регулярности углов

Описание и обоснование

Пусть $P = (p_1, \dots, p_L)^T$ и $Q = (q_1, \dots, q_L)^T$ — два вещественных вектора длины L . Введем меру

$$\tau(P, Q) := \hat{D}(\Theta) = \frac{1}{L-1} \sum_{k=1}^{L-1} (\theta_k - \bar{\theta})^2, \quad (2.26)$$

где $\Theta = (\theta_1, \dots, \theta_L)^T$, $\bar{\theta} = \sum_{k=1}^{L-1} \theta_k / (L-1)$, θ_k — угол между $(p_k, q_k)^T$ и $(p_{k+1}, q_{k+1})^T$, то есть,

$$\theta_k = \arccos \left(\frac{p_k p_{k+1} + q_k q_{k+1}}{\sqrt{p_k^2 + q_k^2} \sqrt{p_{k+1}^2 + q_{k+1}^2}} \right).$$

Заметим, что значения θ_k принадлежат $[0, \pi]$.

Предложение 13. Для сингулярных векторов U_1 и U_2 временного ряда S с элементами, заданными в (1.19), справедливы следующие утверждения.

1. Если $\alpha = 0$ и L целое число, то $\tau(U_1, U_2) = 0$.
2. Если $\alpha = 0$ и $L = [\beta N]$, где $0 < \beta < 1$, то $\lim_{L \rightarrow \infty} \tau(U_1, U_2) = 0$.
3. Если $\alpha = \alpha_N = C/N$, где C — некоторая константа, и $L = [\beta N]$, где $0 < \beta < 1$, то $\lim_{L \rightarrow \infty} \tau(U_1, U_2) = 0$.

Опишем подход к определению колебательной компоненты на основе свойств меры τ .

Равенство $\tau(P, Q) = 0$ означает, что углы между последовательными точками $(p_k, q_k)^T$ равны. Если нормы векторов P и Q монотонно изменяются по k , то двумерные диаграммы P и Q имеют вид спиралей.

Поэтому в условиях предложения 13 для $d = 2$ и достаточно больших L ряд (1.18) можно отождествить с двумерной диаграммой сингулярных векторов U_1 и U_2 и мера τ будет близка к нулю. Примеры результирующих изображений со «спиралями» для рядов с $N = 99$, $L = 50$ представлены на рис. 2.25.

По условиям предложения 13 для $d = 2$, при достаточно больших L ряд (1.18) можно идентифицировать с помощью двумерной диаграммы сингулярных векторов U_1 и U_2 , что соответствует мере τ близкой к нулю. Рис. 2.25 содержит несколько примеров результирующих изображений со «спиралями» для рядов с $N = 99$, $L = 50$.

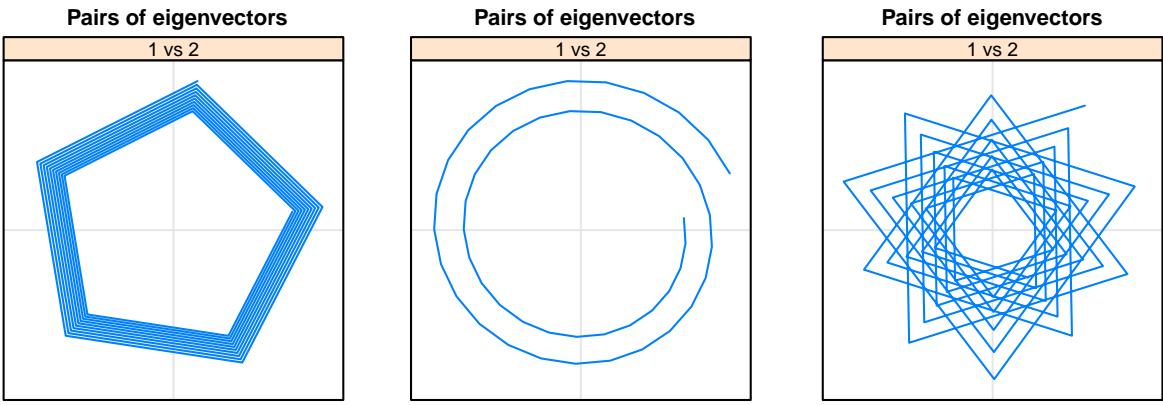


Рис. 2.25: Двумерные диаграммы сингулярных векторов вещественных э.-м. гармоник, $N = 99$, $L = 50$, $\beta = 1/2$.

Заметим, что на практике мы не имеем дело с рядами бесконечной длины, поэтому для ряда конечной длины N с постоянной величиной α условие $\alpha = C/N$ означает не слишком большое значение $e^{\alpha N}$, т.е. ограниченный диапазон значений ряда.

Если известно число э.-м. гармоник, которые мы хотим определить, то алгоритм состоит просто в выборе пар векторов U_i и U_{i+1} с минимальными значениями меры $\tau(U_i, U_{i+1})$, $i = 1, \dots, d-1$.

В случае, когда число э.-м. гармоник неизвестно, пары сингулярных векторов могут быть выбраны с помощью порога, т.е. те векторы U_i и U_{i+1} , значение $\tau(U_i, U_{i+1})$ которых меньше заданного порога, могут быть отнесены к гармонике, но теоретического обоснования выбора порога нет. Поэтому в данном разделе мы приводим эмпирическое обоснование выбора порога и проводим численные исследования на примере зашумленного э.-м. косинуса.

Отметим, что метод не работает для случая $\omega = 0.5$. Поэтому мы не рассматриваем его в этом разделе.

Мы будем называть этот метод *метод идентификации колебательной компоненты по регулярности углов*.

Алгоритм

Все рассуждения, приведенные в предыдущем разделе, объясняли, почему значение меры τ будет равно 0 для сингулярных векторов э.-м. гармоники. Возникает вопрос, можно ли получить значение τ близкое к 0 для сингулярных векторов не э.-м. гармоники.

Например, если некоторые векторы P и Q дают, хотя и разные, малые значения углов θ_k , заданных в определении (2.26) меры τ , то дисперсия $\hat{D}(\Theta)$ из того же определения также будет мала. Чтобы преодолеть эту проблему, мы будем использовать нормированную версию

меры τ для векторов $P = (p_1, \dots, p_L)^T$ и $Q = (q_1, \dots, q_L)^T$:

$$\tilde{\tau}(P, Q) := \frac{\tau(P, Q)}{\min(1, \bar{\theta}^2)} = \frac{\hat{D}(\Theta)}{\min(1, \bar{\theta}^2)} = \frac{1}{(L-1)\min(1, \bar{\theta}^2)} \sum_{k=1}^{L-1} (\theta_k - \bar{\theta})^2, \quad (2.27)$$

где, как и раньше, $\Theta = (\theta_1, \dots, \theta_L)^T$, $\bar{\theta} = \sum_{k=1}^{L-1} \theta_k / (L-1)$ и θ_k — угол между векторами $(p_k, q_k)^T$ и $(p_{k+1}, q_{k+1})^T$.

Деление на $\min(1, \bar{\theta}^2)$ может только увеличить значение меры, поэтому оно борется с проблемой малых значений угла. Малые значения могут принимать углы и для сингулярных векторов э.-м. гармоники, но для них дисперсия будет близка к нулю и нормализация не будет играть существенной роли.

Для $\tilde{\tau}$, как и для τ , утверждение 13 имеет место, так как если $\tau(P, Q) = 0$, то, очевидно, и $\tilde{\tau}(P, Q) = 0$.

Как обсуждалось в предыдущем разделе, метод имеет две модификации: для случая, когда число э.-м. гармоник известно, и для случая, когда число э.-м. гармоник неизвестно. Эти две модификации отличаются критериями остановки в конце алгоритма 2.15, представленными ниже.

АЛГОРИТМ 2.15: 1D-SSA. Метод идентификации колебательной компоненты по регулярности углов

Input: Следующие данные и параметры являются входными.

1. **Данные:** число компонент r ; левые сингулярные векторы $\{U_j\}_{j=1}^r$.
2. **Параметры:** число э.-м. гармоник $m \leq r/2$, или порог $t_0 \geq 0$, в зависимости от критерия остановки.

Output: Группа индексов сингулярных векторов J , связанных с колебательной компонентой.

- 1: На основе $\{U_i\}_{i=1}^r$ вычислить и упорядочить по возрастанию значения $\tilde{\tau}(U_j, U_{j+1})$, $j = 1, \dots, r-1$, используя (2.27). При вычислениях для каждого $j = 2, \dots, r-1$ проверяем: если $\tilde{\tau}(U_j, U_{j+1}) < \tilde{\tau}(U_{j-1}, U_j)$, то отбрасываем $\tilde{\tau}(U_{j-1}, U_j)$ из рассмотрения, иначе отбрасываем $\tilde{\tau}(U_j, U_{j+1})$. В результате получаем значения $\tau_1, \dots, \tau_{[r/2]}$.
- 2: Два варианта критерия остановки следующие. Перебрать элементы множества $i = 1, \dots, [r/2]$.

1. либо до тех пор, пока $i < m$,

2. или пока $\tau_i < t_0$.

Обозначим i_0 момент остановки.

- 3: Множество J состоит из индексов $j, j+1$ сингулярных векторов U_j и U_{j+1} , участвующих в вычислении значений $\tau_1, \dots, \tau_{i_0-1}$.

Замечание 21. Не только r ведущих сингулярных векторов могут быть выходными данными алгоритма. Входными данными может быть любой набор последовательных векторов (набор, в котором индекс каждого следующего вектора на единицу больше предыдущего) из множества $\{U_j\}_{j=1}^d$. В алгоритме для простоты приведен вариант с r ведущими векторами.

Замечание 22. Вопрос о выборе нормализации в определении (2.27) требует дальнейшей проработки. Возможно, если учесть все факторы, нормализацию можно выбрать более подходящую.

Выбор порога

Существует два варианта использования методов автоматической идентификации.

1. Анализ одного временного ряда без предварительной информации.
2. Анализ многих временных рядов схожей структуры (пакетная обработка).

Для второго варианта метод выбора порога прост. Для нескольких рядов из пакета рядов вычисляем значения меры τ для последовательных пар сингулярных векторов; рассматриваем изображения сингулярных векторов, выбираем группу индексов I компонент, принадлежащих колебательной компоненте; выбираем порог t_0 в некоторой точке, между максимальным значением τ для I и минимальным значением τ для остальных индексов $\{1, \dots, d\} \notin I$. Затем мы используем полученное значение порога t_0 для всех оставшихся рядов. В разделе 2.5.4 рассматриваемый метод будет сравниваться с частотным методом именно для этого случая.

Для первого варианта невозможно придумать и теоретически обосновать универсальный метод выбора порога. Поэтому проблема выбора порога, который приводит к извлечению только колебательной компоненты, заменяется проблемой предварительной обработки, которая находит все колебательные компоненты, но может извлечь и что-то другое.

Поэтому, давайте проведем моделирование рядов с одной э.-м. гармоникой и шумом, и посмотрим на 95% квантиль значения $\tau(U_1, U_2)$ для различных уровней шума, а также зависимость этого квантиля от уровня шума.

Далее мы рассматриваем следующие ряды S , N и X для исследования модельных данных:

$$s_k = e^{\alpha k} \cos(2\pi k\omega), \quad (2.28)$$

$$n_k = e^{\alpha k} \sigma \varepsilon_k, \quad (2.29)$$

$$x_k = s_k + n_k, \quad (2.30)$$

где $k = 1, \dots, N$, ε_k — независимые гауссовские случайные величины, распределенные как $N(0, 1)$. Рассмотрим $N = 99$, $L = 50$, $\sigma = 0, 0.2, 0.4, 0.6, 0.8, 1, 1.2, 1.4$.

Для каждого значения σ смоделируем ряд N 1000 раз и по полученной выборке из рядов вида (2.30) подсчитаем 95% квантиль выборочного значения меры $\tau(U_1, U_2)$ от первых двух сингулярных векторов. На рис. 2.26 показан график зависимости квантиля от величины σ .

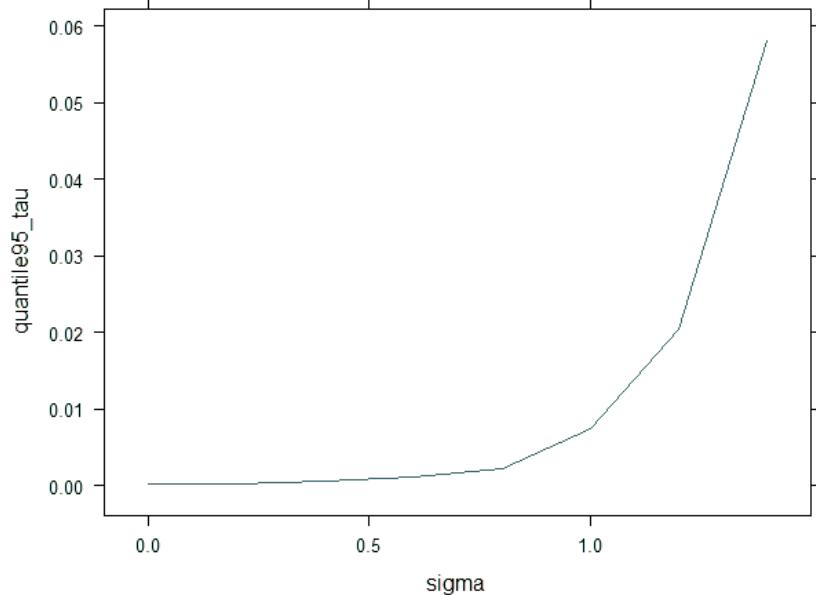


Рис. 2.26: Зависимость 95% квантиля для $\tau(U_1, U_2)$ от величины шума σ .

Значение $\sigma = 1$ очень велико для исследуемой длины ряда, т.е. шум смещивается с колебательной составляющей. Это видно на рис. 2.26: после $\sigma = 1$ график резко пошел вверх. Поэтому стоит рассмотреть результаты для $\sigma \leq 1$. Результаты показывают, что подходящее значение порога составляет примерно $t_0 = 0.01$ для $\sigma = 1$.

2.5.4. Сравнение методов для идентификации гармоники

Сравним метод по регулярности углов с частотным методом из раздела 2.5.2. Как и раньше, мы рассматриваем ряды S , N и X с элементами, заданными (2.28), (2.29) и (2.30).

Цель автоматической идентификации такая же, как и цель визуальной идентификации. Поэтому сравним эти методы следующим образом. Смоделируем э.-м. гармонику S и шум N с $\sigma = 0.2, 0.4, 0.6, 0.8, 1$, $\alpha = 0, 0.01$, $\omega = 1/7$. Как и ранее, $N = 99$ и $L = 50$ (поэтому $L\omega$ не целое число). Для фиксированных значений параметров α , ω , σ будем моделировать ряд $X = S + N$. Восстановление сигнала по двум ведущим собственным тройкам мы будем называть “визуальной идентификацией”.

Пусть $X^{(1)}$ и $X^{(2)}$ — две реализации ряда X . Пусть $S^{(V,1)}$ и $S^{(V,2)}$ — ряды, восстановленные из рядов $X^{(1)}$ и $X^{(2)}$ с помощью «визуальной идентификации».

Обозначим пороговое значение для метода с τ через t_0 , а для частотного метода через ρ_0 . Пусть $S^{(A,1,\varepsilon)}$ и $S^{(A,1,\rho_0)}$ — ряды, восстановленные из ряда $X^{(1)}$ с помощью алгоритмов

автоматической идентификации с пороговыми значениями t_0 и ρ_0 соответственно. Обозначим также множество значений $T = \{0, 0.01, 0.02, \dots, 0.99, 1\}$, т.е. чисел от 0 до 1 с шагом 0.01.

Затем находим значения порогов t_0^{opt} и ρ_0^{opt} путем решения следующих задач минимизации:

$$t_0^{opt} = \arg \min_{t_0 \in T} \left(\frac{1}{N} \sum_{k=0}^{N-1} \left(s_k^{(V,1)} - s_k^{(A,1,t_0)} \right)^2 \right),$$

$$\rho_0^{opt} = \arg \min_{\rho_0 \in T} \left(\frac{1}{N} \sum_{k=0}^{N-1} \left(s_k^{(V,1)} - s_k^{(A,1,\rho_0)} \right)^2 \right).$$

Далее, пусть $S^{(A,2,t_0^{opt})}$ и $S^{(A,2,\rho_0^{opt})}$ — ряды, восстановленные из ряда $X^{(2)}$ с помощью алгоритмов автоматической идентификации с пороговыми значениями t_0^{opt} и ρ_0^{opt} соответственно.

Вычислим ошибки идентификации для этих рядов как

$$E_\tau = \frac{1}{N} \sum_{k=0}^{N-1} \left(s_k^{(V,2)} - s_k^{(A,2,t_0^{opt})} \right)^2, \quad E_\rho = \frac{1}{N} \sum_{k=0}^{N-1} \left(s_k^{(V,2)} - s_k^{(A,2,\rho_0^{opt})} \right)^2.$$

Таким образом, мы вычислили оптимальные пороговые значения для одной реализации ряда X , использовали эти оптимальные пороговые значения для восстановления ряда по второй реализации и вычислили ошибку идентификации для полученного ряда.

Повторим эту процедуру 200 раз и вычислим среднее и медиану ошибок идентификации E_τ и E_ρ . Результаты для $\alpha = 0$, $\omega = 1/7$ приведены в таблице 2.1. Результаты для $\alpha = 0.02$, $\omega = 1/7$ представлены в таблице 2.2.

Таблица 2.1: Сравнение гармонической идентификации; $\alpha = 0$, $\omega = 1/7$.

	среднее τ	среднее ρ	медиана τ	медиана ρ
$\sigma = 0.2$	0.0042	0.040	0	0.0028
$\sigma = 0.4$	0.0156	0.064	0	0.0046
$\sigma = 0.6$	0.0444	0.100	0	0.0342
$\sigma = 0.8$	0.1012	0.151	0	0.0656
$\sigma = 1$	0.1224	0.183	0	0.1179

Таблица 2.2: Сравнение гармонической идентификации; $\alpha = 0.02$, $\omega = 1/7$.

	среднее τ	среднее ρ	медиана τ	медиана ρ
$\sigma = 0.2$	0,0003	0,050	0	0,0010
$\sigma = 0.4$	0,0010	0,080	0	0,0020
$\sigma = 0.6$	0,0082	0,103	0	0,0170
$\sigma = 0.8$	0,0549	0,185	0	0,0655
$\sigma = 1$	0,1917	0,239	0	0,1554

Из таблиц 2.1 и 2.2 можно сделать следующие выводы. Во-первых, для всех значений α и σ , метод идентификации регулярности угла с помощью меры τ дает меньшие ошибки,

чем частотный метод с помощью меры ρ . Во-вторых, оба метода дают примерно одинаковые результаты для немодулированных ($\alpha = 0$) и модулированных гармоник ($\alpha = 0.02$). Для метода идентификации по регулярности углов этот результат подтверждается теорией, так как мера τ не зависит от α . Для частотного метода этот результат неожиданный, так как метод основан на периодограмме сингулярных векторов, а для модулированной гармоники наблюдается “растекания” периодограммы. Результат можно объяснить тем, что для частотного метода алгоритм учитывает возможное растекание, и в качестве ключевой меры ρ , задаваемой формулой (2.23), берется сумма двух соседних значений периодограммы.

Глава 3

Модельные задачи: заполнение пропусков, оценка сигнала, обнаружение сигнала

SSA-анализ временных рядов, который рассматривается в главе 2, может быть классифицирован как свободный от моделей. В этой главе, напротив, мы рассматриваем методологии в рамках подхода 1D-SSA, которые требуют наличия модели. Эти методологии включают такие общие проблемы, как прогнозирование, интерполяция, аппроксимация малого ранга и оценка параметров. Используемая модель основана на свойствах аппроксимирующего подпространства, построенного в процессе 1D-SSA анализа. Глава 2 и методики данной главы относятся к классу основанных на подпространстве методов анализа временных рядов и обработки сигналов.

Как и в главе 2, для краткости, в этой главе мы будем называть 1D-SSA просто SSA. В отличие от SSA-анализа, входными данными для алгоритмов этой главы не обязательно является набор $X_N = (x_1, \dots, x_N)$ из N вещественных чисел; это может быть оценка подпространства. Это подпространство, как правило, получается после шага группировки любого из алгоритмов SSA и имеет вид $\text{span}(U_i, i \in I)$.

Раздел 3.1 посвящен проблеме заполнения недостающих данных во временных рядах.

Раздел 3.2 связан с проблемой оценки сигналов в зашумленных временных рядах, если сигнал имеет малый ранг, а шум является авторегрессией. Существует две формы постановки этой задачи и, соответственно, два подхода к ее решению. Мы рассматриваем конкретную проблему эквивалентности этих форм.

В разделе 3.3 рассматривается задача обнаружения сигнала, где сигнал — синусоида, а шум — красный (т.е. авторегрессионный процесс порядка 1 с положительным коэффициентом).

Глава также содержит раздел 3.4, посвященный выбору параметров в методах, описанных в этой и предыдущих главах.

3.1. Метод SSA заполнения пропусков на основе подпространств

3.1.1. Обзор методов SSA для заполнения пропусков

Этот раздел посвящен расширению алгоритмов прогнозирования SSA для анализа временных рядов с пропущенными данными.

Существует три подхода к решению этой проблемы. Первый подход был предложен в [132]. Этот подход подходит только для стационарных временных рядов и использует следующую простую идею: в процессе вычисления скалярных произведений векторов с отсутствующими компонентами мы используем только пары компонент вектора без пропусков и

опускаем остальные. Пакет RSSA не реализует этот подход ввиду его ограниченности. Поэтому мы сосредоточимся на двух других подходах, основанном на подпространстве [4] и итерационном [133].

Обычно проблема заполнения отсутствующих данных формулируется как проблема заполнения значений сигнала. Однако проблема заполнения является более общей. Например, можно быть заинтересованным в интерполяции недостающих данных в тренде или сезонности. Для этого структура, которая нас интересует, должна быть обнаружена методом. С точки зрения SSA, это означает, что интересующий нас компонент ряда должен быть отделен от остатка, а также правило для выделения компонента должно быть фиксированным (например, индексы собственных троек для восстановления должны быть заданы в Basic SSA). Поэтому имеет смысл объединить рассмотренные методы с модификациями SSA, описанными в главе 2, которые улучшают разделимость.

Для обнаружения структуры до выполнения заполнения пропусков можно применить метод Shaped SSA к ряду, если расположение пропусков позволяет провести разложение (см. раздел 5.2.1). Если Shaped SSA дает неудовлетворительные результаты (например, если количество запаздывающих векторов без пропусков слишком мало и поэтому обнаружение структуры невозможно), то подход на основе подпространства неприменим. Тем не менее, следующая общая техника может быть применена в рамках итерационного подхода: можно добавлять искусственные пропуски и выбирать параметры метода заполнения пропусков, чтобы минимизировать ошибку заполнения.

Подход на основе подпространства

Метод заполнения пропусков на основе подпространств, предложенный в работе [4] (см. также [29, Раздел 3.7] и [30], является расширением алгоритмов прогнозирования SSA. Мы опишем метод, основанный на подпространстве, в соответствии с этими ссылками.

Для прогнозирования последняя координата вектора в выбранном подпространстве может быть однозначно задана как линейная комбинация первых $L - 1$ координат. Этот подход может быть расширен для заполнения набора неизвестных (отсутствующих) координат вектора в виде линейных комбинаций известных координат. Здесь мы используем найденную структуру сигнала (в виде подпространства) для заполнения пропусков. В частном случае, когда отсутствующие значения расположены в конце ряда, проблема заполнения этих значений совпадает с проблемой прогнозирования.

Предположения для заполнения пропусков такие же, как и для прогнозирования; а именно, SSA должен быть в состоянии приближенно разделить интересующий компонент ряда.

Отметим, что заполнение пропусков в отдельных компонентах сигнала может быть выполнено в виде следующей двухэтапной процедуры: сначала мы заполняем пропуски во всем сигнале, а затем разлагаем восстановленный сигнал на нужные компоненты.

Кластеры отсутствующих данных В подходе, основанном на подпространстве, метод заполнения пропусков может быть применен к различным группам отсутствующих данных независимо друг от друга. Чтобы представить такие независимые кластеры, дадим несколько определений.

Определение 10. Для фиксированного L , последовательность недостающих данных временного ряда называется *кластером отсутствующих данных*, если каждые два соседних отсутствующих значения из этой последовательности разделены менее чем L не пропущенных значений и нет пропущенных данных среди L соседей (если они существуют) левого/правого элемента кластера.

Таким образом, группа из не менее чем L последовательных не пропущенных значений ряда разделяет кластеры с пропущенными данными.

Кластер называется *левым/правым*, если его левый/правый элемент находится на расстоянии менее L от левого/правого конца ряда. Кластер называется *непрерывным*, если он не содержит пропущенных данных.

Схема алгоритма Опишем схему алгоритма в форме, реализованной в пакете RSSA.

Предположим, что у нас есть исходный временной ряд $X_N = (x_1, \dots, x_N)$ состоящий из N элементов, часть из которых неизвестна. Опишем схему алгоритма, предполагая, что мы восстанавливаем первую компоненту $X_N^{(1)}$ наблюдаемого ряда $X_N = X_N^{(1)} + X_N^{(2)}$.

Схема метода следующая. Параметрами являются длина окна L и группа I компонентов в разложении SSA. Мы предполагаем, что расположение недостающих данных позволяет применить Shaped SSA для выбранного L . Предлагаются две версии, ‘последовательная’ и ‘одновременная’. Эти версии соответствуют последовательному рекуррентному прогнозированию и одновременному векторному прогнозированию соответственно.

Схема заполнения пропусков на основе подпространств

1. **Shaped SSA.** Для данного ряда Shaped SSA применяется для заданной длины окна L и группы I . Любая модификация описанная в главе 2 и согласующаяся с Shaped SSA, может быть использована. В результате мы получаем восстановленный ряд и набор ортонормированных векторов, образующих базис для аппроксимированного подпространства сигнала.
2. **Определение кластеров недостающих данных.** Все пропуски разбиваются на кластеры. Для последовательной версии каждый кластер преобразуется в непрерывный; то есть, не пропущенные значения внутри кластера заменяются на пропуски.
3. **Прогнозирование.** Для прогнозирования или заполнения нескольких значений, можно использовать два подхода: последовательный и одновременный. Оба эти подхода

могут использовать либо рекуррентный, либо векторный метод прогнозирования. В текущей версии пакета RSSA, последовательный подход использует рекуррентный метод прогнозирования, в то время как одновременный подход использует метод, аналогичный методу векторного прогнозирования.

Последовательное заполнение пропусков делает прогнозирование слева и справа для каждого кластера с последующим взвешенным усреднением результатов прогнозирования; при этом подпространство, используемое для прогнозирования, оценивается методом Shaped SSA. Если кластер левый или правый, то для заполнения пропусков используется только один прогноз.

Для одновременного заполнения пропусков используется так называемое одновременное прогнозирование. В [29, Раздел 3.1], одновременное прогнозирование описывается в его рекуррентной версии. Поскольку здесь мы используем векторную версию, одновременное заполнение пропусков состоит из двух операций, называемых “ Π -проекция” и “одновременное заполнение”.

Обсуждение Отметим, что для успешной интерполяции необходима приближенная отделимость заполняемого компонента. Для точно отделенного компонента недостающие значения могут быть восстановлены без ошибок. Расположение пропущенных данных очень важно для возможности заполнения методом подпространства, так как количество непропущенных значений должно быть достаточно большим для достижения разделимости методом Shaped SSA. По крайней мере, количество полных запаздывающих векторов должно быть больше, чем ранг заполняемого компонента временного ряда.

Алгоритмы Ниже мы приводим лишь краткое описание алгоритмов заполнения на основе подпространств. Алгоритмы могут работать с несколькими кластерами пропусков. Мы описываем их версии только для одного внутреннего кластера.

Следующий алгоритм соответствует комбинации методов ‘последовательное заполнение слева’ и ‘последовательное заполнение справа’. В алгоритмах заполнения пропусков мы будем использовать термин Shaped SSA, который будет введен только в главе 5, однако, в применении к ряду с пропусками Shaped SSA отличается от обычного SSA только тем, что траекторная матрица строится не из всех векторов вложения, а только из тех, в которых нет пропусков.

АЛГОРИТМ 3.1: Последовательное рекуррентное заполнение пропусков на основе подпространств

Input: Временной ряд X длины N , содержащий пропуски, которые начинаются с i -ой точки и заканчиваются в j -ой точке, набор индексов пропусков $P = \{i, \dots, j\}$, $p = |P|$, длина окна L , модификация SSA, группа собственных троек I .

Output: Восстановленный компонент ряда \tilde{X} с заполненными пропусками.

- 1: Применить Shaped SSA выбранной модификации к \tilde{X} (раздел 5.2.1) и получить подпространство $\mathcal{L} = \text{span}\{P_i, i \in I\}$ и восстановленный ряд \tilde{X} с пропусками.
 - 2: Применить прямой рекуррентный алгоритм прогнозирования в подпространстве \mathcal{L} , начиная с $(\tilde{x}_{i-L+1}, \dots, \tilde{x}_{i-1})$, и построить рекуррентный прогноз на p шагов G^{left} .
 - 3: Применить алгоритм обратного рекуррентного прогнозирования в подпространстве \mathcal{L} , начиная с $(\tilde{x}_{j+L-1}, \dots, \tilde{x}_{j+1})$ и построить рекуррентный прогноз на p шагов G^{right} .
 - 4: Объединить G^{left} и G^{right} для получения G . Например, $g_i = (1 - \alpha_i)g_i^{\text{left}} + \alpha_i g_i^{\text{right}}, i = 1, \dots, p$, где $\alpha_i = i/(p+1)$.
 - 5: Задать $\tilde{X}|_P = G$.
-

Заметим, что если пропуск правый (или левый), то применяется только прямое (или обратное) рекуррентное прогнозирование. Отдельного обоснования алгоритм 3.1 не требует, так как основан на алгоритме прогнозирования.

Следующий алгоритм соответствует комбинации метода ‘П-проектора’ и ‘одновременного заполнения’.

АЛГОРИТМ 3.2: Одновременное векторное заполнение пропусков на основе подпространства

Input: Временной ряд X длины N , содержащий пропуски, которые начинаются с i -й точки и заканчиваются в j -ой точке, набор индексов пропусков $P = \{i, \dots, j\}$, $p = |P|$, длина окна L , модификация SSA, группа собственных троек I .

Output: Восстановленный компонент ряда \tilde{X} с заполненным пропуском.

- 1: Применить Shaped SSA выбранной модификации к \tilde{X} (раздел 5.2.1) и получить подпространство $\mathcal{L} = \text{span}\{P_i, i \in I\}$ и восстановленную матрицу \hat{X} , состоящую из векторов с непропущенными значениями во всех позициях.
 - 2: Продолжить значения полных восстановленных векторов в соответствии со структурой ганкелевой матрицы, чтобы получить частично заполненные восстановленные векторы.
 - 3: Спроектировать действительные части векторов с помощью П-проектора.
 - 4: Одновременно заполнить недостающие части векторов. Если это невозможно, то поставьте NA (‘not available’).
 - 5: Ганкелизовать матрицу \hat{X} для получения ряда \tilde{X} . Ганкелизация выполняется путем усреднения по непропущенным значениям. Если нет непропущенных значений, то результатом будет NA.
-

Теоретическое обоснование алгоритма 3.2 содержится в подразделах 3.1.2–3.1.4.

Итеративный подход

Естественной и простой идеей для заполнения пропущенных значений является итерационный подход, когда пропуски сначала заполняются с помощью некоторых разумных значений, а затем эти значения итеративно улучшаются путем обновления приближений SSA для базовой структуры объекта. Эта идея была предложена в [134] для заполнения недостающих значений в зашумленных матриц неполного ранга и позже была распространена на временные ряды в работе [133].

Для матрицы неполного ранга структура определяется ее рангом и поэтому улучшение заполнения осуществляется с помощью SVD, где первые r компонент SVD описывают эту структуру. Для временного ряда конечного ранга r улучшение может быть получено с помощью SVD его траекторной матрицы с последующей ганкелизацией. Заметим, что это именно Basic SSA с восстановлением. Кроме того, Toeplitz SSA или SSA с проекцией могут быть использованы на итерациях, если ряд стационарен или нам частично известна модель ряда.

На каждой итерации мы вставляем улучшенные значения в места пропусков и восстанавливаем исходные данные в местах непропущенных значений.

Описанный выше подход может быть формально применен практически для любого расположения пропущенных значений. Численные эксперименты показывают, что итерационный подход может не сработать, если пропущенные данные расположены на концах временного ряда.

Итерационный подход не имеет строгого доказательства сходимости. Другим недостатком итерационного подхода является его невозможность точно заполнить пропуски даже для бесшумных сигналов. Кроме того, итерационный метод имеет большие вычислительные затраты.

Запишем алгоритм итерационного заполнения пропусков. Для набора \mathbf{Y} и набора индексов P обозначим через $\mathbf{Y}|_P$ часть набора с индексами от $\mathbf{Y}|_P$. набора с индексами из P . Зададим $\mathcal{N} = \{1, \dots, N\}$.

АЛГОРИТМ 3.3: Итеративное заполнение пропусков

Input: Временной ряд \mathbf{X} длины N , содержащий пропуски, набор индексов пропущенных значений P , длина окна L , модификация SSA, ряд \mathbf{G} длины N как источник начальных значений для пропусков, ранг для восстановления r , критерий остановки STOP.

Output: Восстановленная компонента ряда $\tilde{\mathbf{X}}$ без пропусков.

- 1: $k \leftarrow 0$, $\tilde{\mathbf{G}}^{(k)}|_P = \mathbf{G}|_P$, $I = \{1, \dots, r\}$.
- 2: Задать $\tilde{\mathbf{X}}^{(k+1)}$ так, что $\tilde{\mathbf{X}}^{(k+1)}|_{\mathcal{N} \setminus P} = \mathbf{X}|_{\mathcal{N} \setminus P}$ и $\tilde{\mathbf{X}}^{(k+1)}|_P = \mathbf{G}^{(k)}|_P$.
- 3: Применить выбранную модификацию SSA с выбранными L и I к $\tilde{\mathbf{X}}^{(k+1)}$ и получить восстановленный ряд $\mathbf{G}^{(k+1)}$.
- 4: $k \leftarrow k + 1$

5: Если не STOP, перейти к шагу 2; иначе $\tilde{X} = G^{(k)}$.

На вход алгоритма может быть подано несколько групп индексов I_k , $k = 1, \dots, m$. Тогда итерации выполняются для $r = \max\{i : i \in I_k, k = 1, \dots, m\}$. В этом случае восстановление на последнем шаге перед STOP выполняется для каждой группы I_k отдельно.

3.1.2. Вспомогательные утверждения

Начнем теоретическое обоснование алгоритма 3.2 со вспомогательных утверждений.

Восстановление компонент вектора в подпространстве

Давайте введем необходимые обозначения. Определим $\mathcal{I} = \{1, \dots, n\}$ и обозначим через $\mathcal{S} = \{i_1, \dots, i_s\} \subset \mathcal{I}$ упорядоченное множество, $|\mathcal{S}| = s$. Пусть \mathbf{I}_s обозначает единичную $s \times s$ матрицу.

По определению, *сужением вектора* $X = (x_1, \dots, x_n)^T \in \mathbb{R}^n$ на множество индексов \mathcal{S} является вектор $X|_{\mathcal{S}} = (x_{i_1}, \dots, x_{i_s})^T \in \mathbb{R}^{|\mathcal{S}|}$.

Сужение матрицы на множество индексов — это матрица, состоящая из сужений ее векторов=столбцов векторов на это множество.

Сужение q -мерного подпространства \mathfrak{G}_q на множество индексов \mathcal{S} — это подпространство, порожденное сужениями всех векторов \mathfrak{G}_q на это множество; суженное подпространство будем обозначать $\mathfrak{G}_q|_{\mathcal{S}}$. Легко доказать, что для любого базиса $\{H_i\}_{i=1}^q$ подпространства \mathfrak{G}_q равенство $\mathfrak{G}_q|_{\mathcal{S}} = \text{span}(H_1|_{\mathcal{S}}, \dots, H_q|_{\mathcal{S}})$ выполняется.

Рассмотрим m -мерное подпространство $\mathfrak{D}_m \in \mathbb{R}^n$. Обозначим через $\{R_k\}_{k=1}^m$ ортонормированный базис \mathfrak{D}_m и определим матрицу $\mathbf{R} = [R_1 : \dots : R_m]$. Зададим упорядоченное множество индексов \mathcal{P} .

Предложение 14. Пусть матрица $\mathbf{I}_{|\mathcal{P}|} - \mathbf{R}|_{\mathcal{P}}(\mathbf{R}|_{\mathcal{P}})^T$ несингулярна. Тогда для любого вектора $X \in \mathfrak{D}_m$ справедлива следующая формула, выражющая $X|_{\mathcal{P}}$ в терминах $X|_{\mathcal{J} \setminus \mathcal{P}}$:

$$X|_{\mathcal{P}} = (\mathbf{I}_{|\mathcal{P}|} - \mathbf{R}|_{\mathcal{P}}(\mathbf{R}|_{\mathcal{P}})^T)^{-1} \mathbf{R}|_{\mathcal{P}}(\mathbf{R}|_{\mathcal{J} \setminus \mathcal{P}})^T X|_{\mathcal{J} \setminus \mathcal{P}}. \quad (3.1)$$

Доказательство. Для простоты обозначений пусть $\mathcal{P} = \{1, \dots, |\mathcal{P}|\}$. Обозначим $X_1 = X|_{\mathcal{P}}$, $X_2 = X|_{\mathcal{J} \setminus \mathcal{P}}$, $\mathbf{R}_1 = \mathbf{R}|_{\mathcal{P}}$, $\mathbf{R}_2 = \mathbf{R}|_{\mathcal{J} \setminus \mathcal{P}}$. Так как $\mathbf{R}\mathbf{R}^T X = X$ для $X \in \mathfrak{D}_m$ и

$$\mathbf{R}\mathbf{R}^T = \begin{pmatrix} \mathbf{R}_1\mathbf{R}_1^T & \mathbf{R}_1\mathbf{R}_2^T \\ \mathbf{R}_2\mathbf{R}_1^T & \mathbf{R}_2\mathbf{R}_2^T \end{pmatrix},$$

имеем $X_1 = \mathbf{R}_1\mathbf{R}_1^T X_1 + \mathbf{R}_1\mathbf{R}_2^T X_2$. Возвращаясь к первоначальным обозначениям, мы приходим

к равенству

$$(\mathbf{I}_{|\mathcal{P}|} - \mathbf{R}|_{\mathcal{P}} (\mathbf{R}|_{\mathcal{P}})^T) X|_{\mathcal{P}} = \mathbf{R}|_{\mathcal{P}} (\mathbf{R}|_{\mathcal{J} \setminus \mathcal{P}})^T X|_{\mathcal{J} \setminus \mathcal{P}}.$$

На этом доказательство завершено. \square

Лемма 9. Следующие условия эквивалентны:

- 1) $\dim \mathfrak{D}_m|_{\mathcal{J} \setminus \mathcal{P}} = \dim \mathfrak{D}_m$;
- 2) векторы $\{R_k|_{\mathcal{J} \setminus \mathcal{P}}\}_{k=1}^m$ линейно независимы и образуют базис подпространства $\mathfrak{D}_m|_{\mathcal{J} \setminus \mathcal{P}}$;
- 3) $Y|_{\mathcal{J} \setminus \mathcal{P}} \neq 0|_{\mathcal{J} \setminus \mathcal{P}}$ для любого ненулевого вектора $Y \in \mathfrak{D}_m$;
- 4) $\text{span}(\mathbf{e}_i, i \in \mathcal{P}) \cap \mathfrak{D}_m = \{0_n\}$;
- 5) матрица $(\mathbf{I}_{|\mathcal{P}|} - \mathbf{R}|_{\mathcal{P}} (\mathbf{R}|_{\mathcal{P}})^T)^{-1}$ существует.

Доказательство. Эквивалентность условий 1) – 4) очевидна.

Докажем эквивалентность условий 4) и 5). Из предложения 14 вытекает эквивалентность 5) следующему утверждению: для любого вектора $V \in \mathfrak{D}_m|_{\mathcal{J} \setminus \mathcal{P}}$ существует единственный вектор $G \in \mathfrak{D}_m$ такой, что $V = G|_{\mathcal{J} \setminus \mathcal{P}}$. Докажем эквивалентность 4) тому же утверждению.

Так как векторы $\{R_k|_{\mathcal{J} \setminus \mathcal{P}}\}_{k=1}^m$ порождают пространство $\mathfrak{D}_m|_{\mathcal{J} \setminus \mathcal{P}}$, вектор V может быть выражен как $V = \sum_{k=1}^m a_k R_k|_{\mathcal{J} \setminus \mathcal{P}}$. Тогда искомый вектор имеет вид $G = \sum_{k=1}^m a_k R_k$. Предположим, что существуют два различных вектора G_1 и G_2 такие, что $V = G_1|_{\mathcal{J} \setminus \mathcal{P}} = G_2|_{\mathcal{J} \setminus \mathcal{P}}$. Рассмотрим их разность $G_1 - G_2 = \sum_{i \in \mathcal{P}} \alpha_i \mathbf{e}_i \in \mathfrak{D}_m$. Эта разность не равна нулевому вектору тогда и только тогда, когда $\text{span}(\mathbf{e}_i, i \in \mathcal{P}) \cap \mathfrak{D}_m = \{0_n\}$. Лемма доказана. \square

Замечание 23. Из пункта 4) леммы 9 следует, что $n - m \geq |\mathcal{P}|$. Это ограничение на число недостающих компонентов вектора является необходимым условием для применения формулы (3.1).

Рассмотрим два частных случая, когда первая ($\mathcal{P} = \{1\}$) или последняя ($\mathcal{P} = \{n\}$) координата выражается через остальные.

Следствие 6. Обозначим $v^2 = \pi_1^2 + \dots + \pi_m^2$, где π_i – это n -ый компонент вектора R_i , и $\{\underline{R}_i\}_{i=1}^m$ – векторы $\{R_i\}_{i=1}^m$ без последних компонент (их размерность равна $n - 1$). Предположим, что $\mathbf{e}_n \notin \mathfrak{D}_m$ и $X = (x_1, \dots, x_n)^T \in \mathfrak{D}_m$. Тогда $v^2 < 1$ и $x_n = \sum_{k=1}^{n-1} a_k x_{n-k}$, где

$$(a_{n-1}, \dots, a_1)^T = \frac{1}{1 - v^2} \sum_{i=1}^m \pi_i \underline{R}_i. \quad (3.2)$$

Следствие 7. Обозначим $\mu^2 = \rho_1^2 + \dots + \rho_m^2$, где ρ_i – это первая координата вектора R_i , и $\{\overline{R}_i\}_{i=1}^m$ – векторы $\{R_i\}_{i=1}^m$ без первых компонент (их размерность равна $n - 1$). Предполо-

жим, что $\mathbf{e}_1 \notin \mathfrak{D}_m$ и $X = (x_1, \dots, x_n)^T \in \mathfrak{D}_m$. Тогда $\mu^2 < 1$ и $x_1 = \sum_{k=2}^n a_k x_k$, где

$$(a_2, \dots, a_n)^T = \frac{1}{1 - \mu^2} \sum_{i=1}^m \rho_i \bar{R}_i. \quad (3.3)$$

Формула (3.2) — это именно та формула, которая задает коэффициенты для рекуррентного SSA-прогнозирования.

Оператор проекции

Рассмотрим подпространства $\mathcal{L}^{(1)}$ и $\mathcal{L}^{(2)}$ из \mathbb{R}^n , размерностью m и $\tilde{m} \leq n - m$ соответственно. Сначала найдем матрицу оператора $\Pi_{\mathcal{J} \setminus \mathcal{P}}^{(1)}$, соответствующую ортогональной проекции $\mathbb{R}^{|\mathcal{J} \setminus \mathcal{P}|} \rightarrow \mathcal{L}^{(1)}|_{\mathcal{J} \setminus \mathcal{P}}$. Обозначим через $\{R_k\}_{k=1}^m$ ортонормированный базис пространства $\mathcal{L}^{(1)}$, $\mathbf{R} = [R_1 : \dots : R_m]$. Зададим $\mathbf{V} = \mathbf{R}|_{\mathcal{J} \setminus \mathcal{P}}$ и $\mathbf{W} = \mathbf{R}|_{\mathcal{P}}$ для удобства обозначений.

Предложение 15. Предположим, что матрица $\mathbf{I}_{|\mathcal{P}|} - \mathbf{WW}^T$ является несингулярной. Тогда матрица $\Pi_{\mathcal{J} \setminus \mathcal{P}}^{(1)}$ оператора ортогонального проектирования $\Pi_{\mathcal{J} \setminus \mathcal{P}}^{(1)}$ имеет вид

$$\Pi_{\mathcal{J} \setminus \mathcal{P}}^{(1)} = \mathbf{VV}^T + \mathbf{VW}^T(\mathbf{I}_{|\mathcal{P}|} - \mathbf{WW}^T)^{-1}\mathbf{WW}^T. \quad (3.4)$$

Доказательство. Введем матрицу $\mathbf{A} = \mathbf{V}^T \mathbf{V}$ (\mathbf{A} и $\mathbf{W}^T \mathbf{W}$ — матрицы $m \times m$). Согласно лемме 9, матрица \mathbf{A} является несингулярной как матрица Грама линейно независимого набора векторов; поэтому \mathbf{A} является обратимой. Известно, что в этом случае оператор ортогонального проектирования на пространство, охватываемое столбцами матрицы \mathbf{V} имеет вид $\Pi_{\mathcal{J} \setminus \mathcal{P}}^{(1)} = \mathbf{VA}^{-1}\mathbf{V}^T$. Задача состоит в том, чтобы найти явную формулу матрицы \mathbf{A}^{-1} .

Поскольку векторы $\{R_k\}_{k=1}^m$ составляют ортонормированный базис пространства $\mathcal{L}^{(1)}$, имеем $\mathbf{R}^T \mathbf{R} = \mathbf{I}_m$. С другой стороны,

$$\mathbf{R}^T \mathbf{R} = (\mathbf{R}|_{\mathcal{J} \setminus \mathcal{P}})^T \mathbf{R}|_{\mathcal{J} \setminus \mathcal{P}} + (\mathbf{R}|_{\mathcal{P}})^T \mathbf{R}|_{\mathcal{P}} = \mathbf{A} + \mathbf{W}^T \mathbf{W}.$$

Таким образом, $\mathbf{A} = \mathbf{I}_m - \mathbf{W}^T \mathbf{W}$. Прямая демонстрация

$$(\mathbf{I}_m - \mathbf{W}^T \mathbf{W})(\mathbf{I}_m + \mathbf{W}^T(\mathbf{I}_{|\mathcal{P}|} - \mathbf{WW}^T)^{-1}\mathbf{W}) = \mathbf{I}_m$$

завершает доказательство. \square

Обозначим через $\Pi^{(1)}$ оператор ортогонального проектирования \mathbb{R}^n на $\mathcal{L}^{(1)}$. Очевидные условия перестановочности проекции и процедуры сужения приведены в следующем предложении.

Предложение 16. Если $\mathcal{L}^{(1)} \perp \mathcal{L}^{(2)}$ и $\mathcal{L}^{(1)}|_{\mathcal{J} \setminus \mathcal{P}} \perp \mathcal{L}^{(2)}|_{\mathcal{J} \setminus \mathcal{P}}$, тогда

$$\boldsymbol{\Pi}_{\mathcal{J} \setminus \mathcal{P}}^{(1)}(X|_{\mathcal{J} \setminus \mathcal{P}}) = (\boldsymbol{\Pi}^{(1)}X)|_{\mathcal{J} \setminus \mathcal{P}}$$

для любого $X \in \mathcal{L}^{(1)} \oplus \mathcal{L}^{(2)}$.

3.1.3. Запаздывающие векторы и траекторные пространства временных рядов конечного ранга с отсутствующими данными

Восстановление недостающих компонент запаздывающих векторов

Рассмотрим временной ряд X_N конечного ранга d и установим длину окна L , $d < \min(L, K)$, $K = N - L + 1$. Предположим, что некоторые данные отсутствуют в наблюдаемом ряде X_N (т.е. некоторые данные считаются неизвестными), но его L -траекторное пространство $\mathcal{L}_d \subset \mathbb{R}^L$ известно.

Возьмем неполный L -запаздывающий вектор, который содержит как пропущенные, так и непропущенные значение ряда X_N . Обозначим этот вектор через X , а упорядоченное множество индексов его пропущенных компонент — \mathcal{P} . В дальнейшем \mathcal{J} обозначает множество $\{1, \dots, L\}$ и набор индексов $\mathcal{J} \setminus \mathcal{P}$ означает множество непропущенных компонент вектора X . Сужения $X|_{\mathcal{P}}$ и $X|_{\mathcal{J} \setminus \mathcal{P}}$ являются тем самым векторами, которые состоят из пропущенных и непропущенных компонент вектора X соответственно.

Для решения задачи восстановления пропущенных компонент $X|_{\mathcal{P}}$ вектора X по непропущенным компонентам $X|_{\mathcal{J} \setminus \mathcal{P}}$, применяем теорию подраздела 3.1.2 (с $m = d$, $\mathfrak{D}_m = \mathcal{L}_d$) к вектору X . Предположим, что траекторное пространство \mathcal{L}_d удовлетворяет условиям предложения 14. Тогда формула (3.1) решает задачу нахождения $X|_{\mathcal{P}}$ через $X|_{\mathcal{J} \setminus \mathcal{P}}$ и тем самым восстанавливает недостающие значения ряда X_N , которые принадлежат вектору X .

Обратимся к случаю, когда исходный ряд с недостающими данными является суммой двух разделимых рядов конечного ранга $X_N^{(1)}$ и $X_N^{(2)}$, т.е. $X_N = X_N^{(1)} + X_N^{(2)}$. Предположим, что траекторные пространства $\mathcal{L}^{(1)}$ и $\mathcal{L}^{(2)}$ рядов $X_N^{(1)}$ и $X_N^{(2)}$ соответственно известны. Поскольку эти ряды по предположению разделимы, их траекторные пространства ортогональны: $\mathcal{L}^{(1)} \perp \mathcal{L}^{(2)}$. Снова, пусть X будет запаздывающим вектором ряда X_N с индексами отсутствующих данных из \mathcal{P} и $X = X^{(1)} + X^{(2)}$, где $X^{(i)}$, $i = 1, 2$, — это соответствующие запаздывающие векторы рядов $X_N^{(i)}$. Решим задачу нахождения запаздывающего вектора $X^{(1)}$ первого ряда через $X|_{\mathcal{J} \setminus \mathcal{P}}$.

Задача распадается на две: нахождение $X^{(1)}|_{\mathcal{J} \setminus \mathcal{P}}$ и нахождение $X^{(1)}|_{\mathcal{P}}$. Если первая задача решена, мы можем применить теорию подраздела 3.1.2 с $m = \text{rank } X_N^{(1)}$ и $\mathfrak{D}_m = \mathcal{L}^{(1)}$, чтобы выразить $X^{(1)}|_{\mathcal{P}}$ через $X^{(1)}|_{\mathcal{J} \setminus \mathcal{P}}$. Предложения раздела 3.1.2 дают нам решение проблемы нахождения $X^{(1)}|_{\mathcal{J} \setminus \mathcal{P}}$, если выполняется условие ортогональности суженных траекторных пространств $\mathcal{L}^{(1)}|_{\mathcal{J} \setminus \mathcal{P}}$ и $\mathcal{L}^{(2)}|_{\mathcal{J} \setminus \mathcal{P}}$. Так как подпространства $\mathcal{L}^{(1)}$ и $\mathcal{L}^{(2)}$ ортогональны, то запаздывающий вектор $X^{(1)}$ ряда $X_N^{(1)}$ равен $\boldsymbol{\Pi}^{(1)}X$; поэтому соответствующие члены ряда $X_N^{(1)}$

можно получить ортогональным проектированием вектора $X|_{\mathcal{I} \setminus \mathcal{P}}$ на $\mathcal{L}^{(1)}|_{\mathcal{I} \setminus \mathcal{P}}$.

Таким образом, $X^{(1)}|_{\mathcal{I} \setminus \mathcal{P}} = \Pi_{\mathcal{I} \setminus \mathcal{P}}^{(1)} X|_{\mathcal{I} \setminus \mathcal{P}}$, где $\Pi_{\mathcal{I} \setminus \mathcal{P}}^{(1)}$ определяется по формуле (3.4) с матрицей \mathbf{R} , столбцы которой являются векторами ортонормированного базиса $\mathcal{L}^{(1)}$, а $X^{(1)}|_{\mathcal{P}}$ выражается через $X^{(1)}|_{\mathcal{I} \setminus \mathcal{P}}$ по формуле (3.1) с той же матрицей \mathbf{R} . Таким образом, значения ряда $X_N^{(1)}$, принадлежащие его запаздывающему вектору $X^{(1)}$, найдены, включая те, которые находятся на местах отсутствующих данных ряда X_N .

3.1.4. Поиск траекторных пространств исходного временного ряда и его аддитивных компонентов

Рассмотрим временной ряд X_N с $\text{rank}_L(X_N) = d$ и его L -запаздывающие векторы $\{X_i\}_{i=1}^K$. Сначала получим условия возможности нахождения базиса траекторного пространства $\mathcal{L}_d = \text{span}(X_i, i = 1, \dots, K)$, используя только непропущенные значения наблюдаемого ряда. Обозначим через $\mathcal{C} \subset \{1, \dots, K\}$ множество номеров полных запаздывающих векторов без пропусков. Предположим, что $\mathcal{C} \neq \emptyset$, и рассмотрим матрицу $\tilde{\mathbf{X}}$, состоящую из запаздывающих векторов $X_i, i \in \mathcal{C}$, в качестве своих столбцов. Пусть $\tilde{\mathcal{L}}_d = \text{span}(X_i, i \in \mathcal{C})$. Легко доказать следующее утверждение.

Предложение 17. *Множество $X_i, i \in \mathcal{C}$, содержит не менее d линейно независимых векторов если и только если $\tilde{\mathcal{L}}_d = \mathcal{L}_d$. В этом случае собственные векторы U_1, \dots, U_d матрицы $\tilde{\mathbf{S}} = \tilde{\mathbf{X}}\tilde{\mathbf{X}}^T$, соответствующие d ненулевым собственным значениям $\tilde{\mathbf{S}}$, образуют ортонормированный базис подпространства \mathcal{L}_d .*

Сформулируем более конструктивное достаточное условие.

Предложение 18. *Если временной ряд X_N имеет L -ранг d , $\mathbf{e}_1 \notin \mathcal{L}_d$, $\mathbf{e}_L \notin \mathcal{L}_d$ и ряд содержит по крайней мере $L + d - 1$ последовательных непропущенных значений, то $\tilde{\mathcal{L}}_d = \mathcal{L}_d$.*

Доказательство. Чтобы доказать это предложение, покажем, что существует d линейно независимых векторов среди полных векторов с запаздыванием. Из условий этого предложения следует, что существует такое число k , что запаздывающие векторы $\{X_{k+i}\}_{i=1}^d$ не содержат пропущенных данных.

Поскольку $\mathbf{e}_1 \notin \mathcal{L}_d$ и $\mathbf{e}_L \notin \mathcal{L}_d$, временной ряд X_N может быть продолжен до бесконечного в обоих направлениях временного ряда X L -ранга d (доказательство основано на следствии 6 и следствии 7 с $n = L$, $m = d$, $\mathfrak{D}_m = \mathcal{L}_d$). Отсюда следует, что X является временным рядом конечного ранга d и, в частности, $\text{rank}_{d+1} X_N = d$, $\mathbf{e}_1, \mathbf{e}_{d+1} \notin \mathcal{L}^{(d+1)}(X_N)$. Следовательно, в силу следствия 6 при $n = d + 1$, $m = d$, $\mathfrak{D}_m = \mathcal{L}^{(d+1)}(X_N)$,

$$X_{i+d} = \sum_{j=1}^d a_j X_{i+d-j}, \quad 1 \leq i \leq K - d.$$

Следовательно, все запаздывающие векторы с индексами, превышающими $k + d$, могут быть выражены в виде линейной комбинации векторов X_{k+1}, \dots, X_{k+d} .

Аналогично, принимая во внимание следствие 7, мы обнаруживаем, что запаздывающие векторы с индексами от 1 до k могут быть записаны как линейные комбинации векторов X_{k+1}, \dots, X_{k+d} . Таким образом, все запаздывающие векторы выражаются через набор векторов $\{X_{k+i}\}_{i=1}^d$. Поскольку размерность траекторного пространства равна d , эти векторы независимы. \square

Теперь мы перейдем к случаю, когда наблюдаемый ряд является суммой двух разделимых рядов: $X_N = X_N^{(1)} + X_N^{(2)}$. Предположим, что условия предложения 17 выполняются и базис траекторного пространства \mathcal{L}_d ряда X_N найден. Сформулируем задачу нахождения траекторного пространства $\mathcal{L}^{(1)}$ и $\mathcal{L}^{(2)}$ ряда $X_N^{(1)}$ и $X_N^{(2)}$, используя $\tilde{\mathbf{X}} = \tilde{\mathbf{X}}^{(1)} + \tilde{\mathbf{X}}^{(2)}$.

Если пространства строк траекторных матриц $\tilde{\mathbf{X}}^{(1)}$ и $\tilde{\mathbf{X}}^{(2)}$ ортогональны, как и пространства столбцов, тогда собственные векторы U_1, \dots, U_d из $\tilde{\mathbf{S}}$, образующие базис \mathcal{L}_d , могут быть в общем случае разбиты на две группы, составляющие базисы пространств $\mathcal{L}^{(1)}$ и $\mathcal{L}^{(2)}$. Траекторные пространства столбцов ортогональны в силу разделимости рядов $X_N^{(1)}$ и $X_N^{(2)}$. Легко показать, что следующие условия являются необходимыми и достаточными для ортогональности траекторных пространств рядов:

$$\sum_{k \in \mathcal{C}} x_{i+k-1}^{(1)} x_{j+k-1}^{(2)} = 0, \quad i, j = 1, \dots, L. \quad (3.5)$$

Таким образом, (3.5) и условия предложения 17 достаточны для нахождения базисов траекторных пространств $\mathcal{L}^{(1)}$ и $\mathcal{L}^{(2)}$.

В качестве примера можно привести сумму константы и гармонического ряда с периодом T . Известно, что если L и K кратны T , то ряды являются разделимыми. Если последовательные значения T отсутствуют, а N и L достаточно велики, то все равно можно найти базисы пространств $\mathcal{L}^{(1)}$ и $\mathcal{L}^{(2)}$.

Условия ортогональности суженных подпространств

Свойства оператора ортогональной проекции $\Pi_{\mathcal{J} \setminus \mathcal{P}}^{(1)}$ были получены в предложении 16 раздела 3.1.2 (и были использованы в разделе 3.1.3). В упомянутом предложении вектор X и множество \mathcal{P} индексов недостающих компонент вектора рассматривались при условии, что пространства $\mathcal{L}^{(1)}|_{\mathcal{J} \setminus \mathcal{P}}$ и $\mathcal{L}^{(2)}|_{\mathcal{J} \setminus \mathcal{P}}$ ортогональны.

Пусть $\mathcal{L}^{(1)}$ и $\mathcal{L}^{(2)}$ — траекторные пространства ряда $X_N^{(1)}$ и $X_N^{(2)}$. Следующие равенства дают нам необходимое и достаточное условие ортогональности $\mathcal{L}^{(1)}|_{\mathcal{J} \setminus \mathcal{P}} \perp \mathcal{L}^{(2)}|_{\mathcal{J} \setminus \mathcal{P}}$:

$$\sum_{k \in \mathcal{J} \setminus \mathcal{P}} x_{i+k-1}^{(1)} x_{j+k-1}^{(2)} = 0, \quad i, j = 1, \dots, K. \quad (3.6)$$

Замечание 24. Если ряды $X_N^{(1)}$ и $X_N^{(2)}$ разделимы, то условие (3.6) эквивалентно

$$\sum_{k \in \mathcal{P}} x_{i+k-1}^{(1)} x_{j+k-1}^{(2)} = 0, \quad i, j = 1, \dots, K. \quad (3.7)$$

Сумма гармонического ряда с периодом T и постоянного ряда, где L, K и число последовательных недостающих данных кратно T , является примером, когда выполняется условие (3.7). Также, L должно быть больше, чем число недостающих значений временного ряда.

3.1.5. Комментарии к реализации подпространственного метода заполнения пропусков

В этом разделе мы опишем детали реализации алгоритмов 3.1 и 3.2.

Методы восстановления значений в позициях непропущенных компонент

Опишем возможные решения задачи по вычислению $\widehat{X}_i|_{\mathcal{J} \setminus \mathcal{P}}, l \leq i \leq p$.

Метод, когда оператор ортогонального проектирования на $\mathcal{M}_r|_{\mathcal{J} \setminus \mathcal{P}}$ применяется к $X_i|_{\mathcal{J} \setminus \mathcal{P}}$ для каждого \mathcal{P} -неполного запаздывающего вектора из кластера пропущенных данных называется “П Проектор”. Предложение 15 при $m = r, \mathcal{L}^{(1)} = \mathcal{M}_r$, и матрица \mathbf{R} , столбцами которой являются собственные векторы $U_i, i \in I_r$, обеспечивают условия применения этого метода и формулы для вычисления $\Pi_{\mathcal{J} \setminus \mathcal{P}}^{(1)} \text{в } \widehat{X}_i|_{\mathcal{J} \setminus \mathcal{P}} = \Pi_{\mathcal{J} \setminus \mathcal{P}}^{(1)} X_i|_{\mathcal{J} \setminus \mathcal{P}}$.

Еще один метод может быть применен к непрерывным (крайним или внутренним) кластерам. Этот метод основан на том, что значения диагональных элементов траекторных матриц (напомним, что $\widehat{\mathbf{X}}$ является аппроксимацией траекторной матрицы ряда $X_N^{(1)}$) с индексами $(i, j), i + j = \text{Const}$, равны. Рассмотрим сначала внутренний кластер. Позиции непропущенных компонент $\widehat{X}_q|_{\mathcal{J} \setminus \mathcal{P}}$ в соответствующем наборе векторов $\widehat{X}_q, q = l, \dots, p$, образуют два “треугольника” (правый и левый). Обозначив через $\widehat{x}_{i,j}$ i -ый элемент вектора \widehat{X}_j , опишем метод на примере заполнения левого “треугольника”, который будет выражаться как множество s -ых “диагоналей”: $\{(i, j) : i + j = s, l + 1 \leq s \leq l + L - 1, l \leq j \leq l + L - 2\}$. Заметим, что “треугольник” и “диагональ” интерпретируются как части матрицы $[\widehat{X}_l : \dots : \widehat{X}_p]$.

Поскольку мы имеем дело с внутренним кластером, существует $l_0, l_0 \geq 1$, левых смежных векторов $\widehat{X}_{l-m}, m = 1, \dots, l_0$, которые являются полными и все их компоненты уже были вычислены на предыдущем шаге. Если мы рассмотрим матрицу $[\widehat{X}_{l-l_0} : \dots : \widehat{X}_p]$, то метод, который можно назвать “компоненты смежных векторов” состоит в замене всех компонент s -ой диагонали $\widehat{x}_{i,j}$ при $i + j = s$ и $l - l_0 \leq j \leq l + L - 2$ на среднее значение $\widehat{x}_{i,j}$ при $i + j = s, l - l_0 \leq j \leq l - 1$. Заметим, что метод также изменяет некоторые компоненты векторов $\widehat{X}_{l-l_0}, \dots : \widehat{X}_{l-1}$, так как использует тот же подход, что и при векторном SSA прогнозировании.

Полностью аналогичная процедура проводится с правым “треугольником”. Для экстремальных кластеров необходимо заполнить только один из “треугольников”.

Методы восстановления значений на местах отсутствующих компонент

Здесь мы предлагаем несколько решений проблемы вычисления $\widehat{X}_i|_{\mathcal{P}}$, $l \leq i \leq p$, если векторы $\widehat{X}_i|_{\mathcal{J} \setminus \mathcal{P}}$ уже получены.

Для первого метода мы потребуем, чтобы каждый вектор из набора запаздывающих векторов удовлетворял условиям предложения 14. Если все условия выполнены, то мы можем восстановить недостающие компоненты каждого вектора по формуле (3.1), где $m = r$, $\mathfrak{D}_m = \mathcal{M}_r$ и \mathbf{R} — матрица, столбцами которой являются собственные векторы $U_i, i \in I_r$. Такой метод восстановления недостающих данных можно назвать “одновременное заполнение”. Заметим, что условия применения одновременного заполнения являются достаточно ограничительными. В частности, невозможно применить метод, если отсутствуют непропущенные компонент хотя бы в одном векторе из набора запаздывающих векторов.

Другие методы основаны на том, что значения с индексами (i, j) на диагоналях $i + j = \text{Const}$ из траекторных матриц равны. Поэтому мы можем восстановить недостающие компоненты одного из запаздывающих векторов и использовать полученные значения для восполнения недостающих элементов соседних векторов.

Рассмотрим набор запаздывающих векторов X_l, \dots, X_p рассматриваемого кластера недостающих значений. Мы предполагаем, что векторы $\widehat{X}_q|_{\mathcal{J} \setminus \mathcal{P}}$, $q = l, \dots, p$, уже были вычислены ранее. Опишем несколько вариантов методов последовательного заполнения.

1) Пусть рассматриваемый кластер не является левым кластером. В этом случае левый вектор X_l содержит только один пропущенный элемент, который находится на месте последней координаты, поэтому набор индексов его недостающих элементов имеет вид $\mathcal{P} = \{L\}$. Следовательно, может быть применено следствие 6. Если $\mathbf{e}_L \notin \mathcal{M}_r$, то мы можем получить последнее значение у вектора \widehat{X}_l как линейную комбинацию предыдущих значений с коэффициентами, заданными формулой (3.2). Далее это значение у используется для заполнения $(L - j)$ -й компоненты векторов \widehat{X}_{l+j} , $j = 1, \dots, L - 1$. После этой процедуры смежный вектор \widehat{X}_{l+1} будет иметь только один недостающий элемент на месте последней координаты (если кластер является непрерывным). Поэтому вся процедура может быть повторена еще раз. И так далее. Такой метод будет называться *последовательное заполнение слева*. Заметим что если кластер недостающих данных не является непрерывным, то процедуру заполнения не нужно применять к некоторым из векторов X_q , $l < q < p$.

2) Последовательное *заполнение справа* полностью аналогично последовательному заполнению слева и сводится к заполнению первой координаты с помощью формулы (3.3), приведенной в следствие 7. Таким образом, условия применения этого метода следующие: кластер недостающих данных не является правым кластером и $\mathbf{e}_1 \notin \mathcal{M}_r$.

3) Различные комбинации последовательного заполнения слева и справа (так называемый двусторонний метод).

Обратим внимание, что условия последовательного заполнения менее строгие по сравнению с одновременной интерполяцией; тем не менее, ошибки могут накапливаться.

Замечание 25. Рассмотрим непрерывный кластер недостающих данных длиной m , который является крайним правым кластером (и нет других кластеров отсутствующих данных в ряде). Если методы “Компоненты смежных векторов” и “Последовательное заполнение слева” будут применены к этому кластеру, то результат совпадет с рекуррентным прогнозом на m шагов вперед; прогнозируемый компонент временного ряда извлекается с помощью Basic SSA из временного ряда, состоящего из первых $N - m$ точек.

Пример

Чтобы продемонстрировать работу методов восполнения недостающих данных, рассмотрим известный временной ряд длины 144, представляющий собой ежемесячное количество пассажиров (в тысячах) на международных авиакомпаниях, начиная с января 1949 года (данные были впервые опубликованы в [135]).

Давайте уберем 12 известных значений, начиная с 68-й точки (т.е. мы считаем, что значения неизвестны за год, начиная с августа 1954 года). Для такого рода искусственно отсутствующих данных мы можем оценить точность их восстановления для различных версий алгоритма. Также для имитации прогноза мы добавляем 12 недостающих данных после последней, 144-й точки ряда. Полученный временной ряд показан на рис. 3.1.

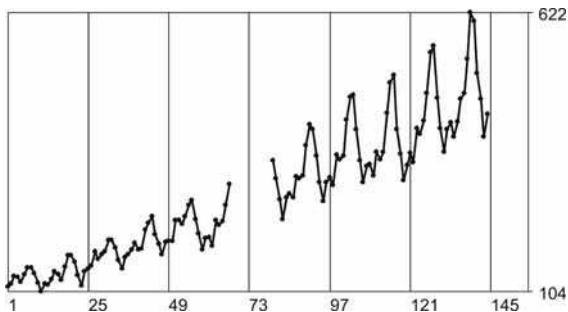


Рис. 3.1: Исходный временной ряд с недостающими данными

Первый вопрос — как выбрать длину окна L . В случае отсутствия недостающих данных общая рекомендация состоит в том, чтобы выбрать длину окна, близкую к половине длины ряда и кратной периоду ожидаемой периодичности (в данном случае 12 месяцев). Длина окна, равная 72, удовлетворяет этим условиям. Однако при таком выборе L все запаздывающие векторы будут содержать пропущенные данные. Поэтому выберем меньшую длину. Выбор $L = 36$ дает нам 62 полных запаздывающих вектора без пропущенных данных.

Анализ собственных векторов U_1, \dots, U_L (для реальных рядов обычно $d = L$) показывает, что собственные векторы с индексами 1, 6 и 9 соответствуют тренду, а собственные векторы с индексами индексами 2–5, 7–8 и 10–13 соответствуют сезонной составляющей. Все остальные собственные векторы могут быть классифицированы как порожденные шумом. Поэтому

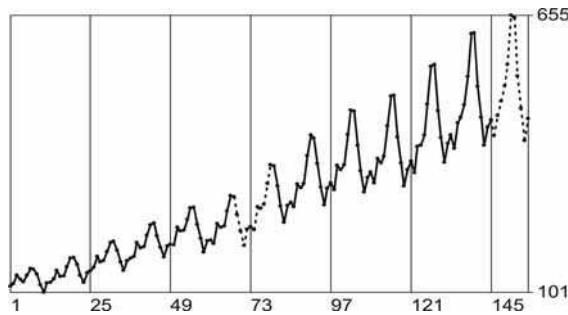


Рис. 3.2: Восстановленный временной ряд с заполненными данными

выберем $r = 13$ и $I_r = \{1, 2, \dots, 13\}$ для того, чтобы восстановить детерминированную составляющую ряда (сигнал).

Все восемь вариантов, описанные выше, применимы к первому непрерывному внутреннему кластеру недостающих данных: два варианта для получения значений на местах непропущенных наблюдений (“П Проектор” и “Компоненты смежных векторов”), и четыре варианта реализации заполнения пропусков (одновременное заполнение и три типа последовательного заполнения). Сравнение результатов восстановления со значениями, которые были искусственно удалены из исходного временного ряда, показывает преимущество варианта “П Проектор” с одновременным заполнением недостающих данных. Ошибка восстановления при этом примерно равна 6 для недостающих данных и равна 4.75 (это ненамного меньше) для других условий.

Применим тот же метод для заполнения второго кластера недостающих данных. Результат показан на рис. 3.2. Восстановленный ряд отмечен пунктирной линией в области пропущенных данных.

3.2. Возможность построения весов в задаче HSLRA

В этом разделе, в соответствии с обозначениями, принятыми в задачах деконволюции, мы будем нумеровать координаты векторов от 0 и обозначать векторы жирными строчными буквами.

Нашей мотивацией является задача выделения сигнала $\mathbf{s} = (s_0, s_1, \dots, s_{N-1})^T$ из наблюдаемого зашумленного сигнала $\mathbf{x} = (x_0, x_1, \dots, x_{N-1})^T = \mathbf{s} + \boldsymbol{\xi}$ длины N , где $\boldsymbol{\xi} = (\xi_0, \xi_1, \dots, \xi_{N-1})^T$ — вектор (ненаблюденного) случайного шума с нулевым средним и ковариационной матрицей $\boldsymbol{\Sigma} = E[\boldsymbol{\xi}\boldsymbol{\xi}^T]$.

Как и ранее, рассматривается модель сигнала в форме (1). Этот класс сигналов может быть определен через низкоранговые ганкелевы матрицы следующим образом. Задаем длину окна L , $1 < L \leq N/2$; $K = N - L + 1$. С рядом $\mathbf{z} = (z_0, z_1, \dots, z_{N-1})^T$ мы связываем траекторную

матрицу

$$\mathcal{T}_L(\mathbf{z}) = \begin{pmatrix} z_0 & z_1 & \dots & z_{K-1} \\ z_1 & \dots & \dots & z_K \\ \dots & \dots & \dots & \dots \\ z_{L-1} & z_L & \dots & z_{N-1} \end{pmatrix} \in \mathbb{R}^{L \times K}.$$

Проблема выделения сигнала конечного ранга может быть сведена к проблеме аппроксимации L -траекторной матрицы $\mathcal{T}_L(\mathbf{x})$ наблюдаемого временного ряда \mathbf{x} ганкелевой матрицей ранга r , т.е., к задаче HSLRA [136, 2, 137].

Задача HSLRA может быть поставлена в двух формах: (а) векторной и (б) матричной. Векторная (в терминах временных рядов) форма этой задачи такова: для заданного $\mathbf{x} \in \mathbb{R}^N$ и положительного целого числа $r < \lfloor N/2 \rfloor$

$$\|\mathbf{x} - \mathbf{y}\|_{\mathbf{W}}^2 \rightarrow \min_{\mathbf{y}: \text{rank } \mathbf{y} \leq r}, \quad (3.8)$$

где $\mathbf{y} = (y_0, y_1, \dots, y_{N-1})^T$, $\|\mathbf{z}\|_{\mathbf{W}}^2 = \mathbf{z}^T \mathbf{W} \mathbf{z}$ для $\mathbf{z} \in \mathbb{R}^N$ и \mathbf{W} — некоторая положительно определенная матрица размера $N \times N$.

Решение (3.8) можно рассматривать как взвешенную оценку наименьших квадратов (WLSE) сигнала \mathbf{s} . Если шум $\boldsymbol{\xi}$ является гауссовским с ковариационной матрицей $\boldsymbol{\Sigma}$, то WLSE с $\mathbf{W}_0 = \boldsymbol{\Sigma}^{-1}$ является оценкой максимального правдоподобия (MLE). Если известны свойства шумового процесса, то векторная форма (3.8) является наиболее естественным способом определения задачи HSLRA. В работе [22] предложен быстрый и устойчивый алгоритм MGN (Modified Gauss-Newton) для решения (3.8) на основе подхода, разработанного в [24], и показано, что MGN превосходит существующий алгоритм, представленный в [138].

Хотя решение задачи в векторной форме имеет быструю реализацию, эти алгоритмы довольно сложны и требуют начальной точки, близкой к решению [139, 138].

Матричная форма задачи HSLRA позволяет использовать простые методы попеременные проекции на подпространства (например, итерации Кэдзоу (1.22)) и, следовательно, с вычислительной точки зрения гораздо предпочтительнее, чем векторная форма (3.8); подробности см. в [30, Раздел 3.4]. Итерации Кэдзоу связаны с SSA тем, что SSA формально можно рассматривать как одну итерацию Кэдзоу.

Определим скалярное произведение в $\mathbb{R}^{L \times K}$ как $\langle \mathbf{X}, \mathbf{Y} \rangle = \langle \mathbf{X}, \mathbf{Y} \rangle_{\mathbf{A}, \mathbf{B}} = \text{tr}(\mathbf{A} \mathbf{X} \mathbf{B} \mathbf{Y}^T)$, где $\mathbf{A} = (a_{i,j})_{i,j=0}^{L-1} \in \mathbb{R}^{L \times L}$, $\mathbf{B} = (b_{i,j})_{i,j=0}^{K-1} \in \mathbb{R}^{K \times K}$; $\|\mathbf{X}\|_{\mathbf{A}, \mathbf{B}}$ — соответствующая норма матрицы в $\mathbb{R}^{L \times K}$. Задача HSLRA в матричной форме представляет собой следующую оптимизационную задачу:

$$\|\mathbf{X} - \mathbf{Y}\|_{\mathbf{A}, \mathbf{B}}^2 \rightarrow \min_{\mathbf{Y} \in \mathcal{M}_r \cap \mathcal{H}}, \quad (3.9)$$

где $\mathcal{H} \subset \mathbb{R}^{L \times K}$ — пространство ганкелевых матриц размера $L \times K$, $\mathcal{M}_r \subset \mathbb{R}^{L \times K}$ — множество матриц ранга не больше r . Для переформулировки исходной задачи HSLRA (3.8) в матрич-

ную форму (3.9), мы должны выбрать $\mathbf{X} = \mathcal{T}_L(\mathbf{x})$ и $\mathbf{Y} = \mathcal{T}_L(\mathbf{y})$; оставшийся вопрос заключается в согласовании векторной нормы в (3.8) с матричной нормой в (3.9).

В работе [23] предложено использовать смешанные попеременные проекции для решения задачи в виде (3.9), где сначала рассматриваются весовые матрицы, соответствующие более медленной сходимости к более точному решению задачи (3.8), а потом — более быстрые, но к менее точному решению. Оказывается, что при этом удается получить более точное и более быстрое решение. В [140] представлен численный алгоритм поиска приближенных матричных весов, который дает субоптимальные веса при заданной степени вырожденности весовых матриц, задаваемой параметром α , $0 < \alpha \leq 1$. Этот алгоритм используется при построении смешанных попеременных проекций в [23], так как параметр α позволяет улучшать точность решения задачи (3.8), при этом замедляя сходимость; и наоборот.

Далее в этом разделе рассматривается решение задачи HSLRA в матричной форме (3.9), которая удобна для нахождения решения с помощью итераций Кэдзоу (1.22). А именно, будет показано, что выбор весов в (3.9), обеспечивающих в точности оптимальное решение, невозможен для случая авторегрессионного шума [19]. В частности, поэтому применение методов численного поиска субоптимальных весов оправдано.

Начнем со вспомогательных результатов.

3.2.1. Свертка матриц и векторов

Пусть A и B — целые положительные числа и $\mathbf{a} = (a_0, \dots, a_A)^T \in \mathbb{R}^{A+1}$ и $\mathbf{b} = (b_0, \dots, b_B)^T \in \mathbb{R}^{B+1}$ — два вектора. Сверткой векторов \mathbf{a} и \mathbf{b} является вектор $\mathbf{c} = \mathbf{a} * \mathbf{b} = (c_0, \dots, c_C)^T \in \mathbb{R}^{C+1}$, где $C = A + B$ и $c_i = \sum_k a_k b_{i-k}$, где $i = 0, 1, \dots, C$ и сумма берется по множеству индексов k таких, что элементы a_k и b_{i-k} определены (то есть, $\max\{0, i-B\} \leq k \leq \min\{A, i\}$).

Определение свертки векторов естественным образом распространяется на матрицы следующим образом. Пусть $\mathbf{A} = (a_{i,j})_{i,j=0}^A$ и $\mathbf{B} = (b_{i,j})_{i,j=0}^B$ — матрицы размеров $(A+1) \times (A+1)$ и $(B+1) \times (B+1)$, соответственно. Матрица $\mathbf{C} = (c_{i,j})_{i,j=0}^{A+B}$ является сверткой матриц \mathbf{A} и \mathbf{B} , если $c_{i,j} = \sum_{k,l} a_{k,l} b_{i-k, j-l}$, где сумма берется по множеству индексов k и l таких, что элементы $a_{k,l}$ и $b_{i-k, j-l}$ определены; то есть, $\max\{0, i-B\} \leq k \leq \min\{A, i\}$ и $\max\{0, j-B\} \leq l \leq \min\{A, j\}$.

Производящая функция (generating function, gf) вектора $\mathbf{u} = (u_0, \dots, u_M)^T$ определяется как $G_{\mathbf{u}}(t) = u_0 + u_1 t + \dots + u_M t^M$. Аналогично, производящая функция матрицы $\mathbf{U} = (u_{i,j})_{i,j=0}^M$ — это $G_{\mathbf{U}}(t, s) = \sum_{i,j=0}^M u_{i,j} t^i s^j$, где $t, s \in \mathbb{C}$.

Из определения свертки следует, что $\mathbf{c} = \mathbf{a} * \mathbf{b}$ тогда и только тогда, когда $G_{\mathbf{c}}(t) = G_{\mathbf{a}}(t)G_{\mathbf{b}}(t)$ для всех $t \in \mathbb{C}$. Аналогично, $\mathbf{C} = \mathbf{A} * \mathbf{B}$ тогда и только тогда, когда $G_{\mathbf{C}}(t, s) = G_{\mathbf{A}}(t, s)G_{\mathbf{B}}(t, s)$ для всех $t, s \in \mathbb{C}$. Из этого следует, что все утверждения о свертках и деконволюциях векторов и матриц могут быть эквивалентно сформулированы на языке производящих функций.

Говорят, что вектор \mathbf{c} может быть слепо деконволютирован, если существуют векторы \mathbf{a} и \mathbf{b} такие, что $\mathbf{c} = \mathbf{a} * \mathbf{b}$. Эта задача (интересная только при некоторых ограничениях на \mathbf{a}

и **b)** эквивалентна изучению корней производящей функции $G_{\mathbf{c}}(t)$.

Матрица \mathbf{C} может быть слепо деконволютирована, если существуют матрицы \mathbf{A} и \mathbf{B} такие, что $\mathbf{C} = \mathbf{A} * \mathbf{B}$.

Существует обширная литература, связанная со сверткой матриц и (слепой) деконволюцией, но, в основном, она связана с приложениями в обработке изображений; см. например, [141, 142]. Нас интересует слепая деконволюция матриц \mathbf{C} , которые пропорциональны обратным к матрицам автоковариации авторегрессионных процессов (ARs).

3.2.2. Связь между векторной и матричной формами HSLRA

Частные случаи соответствия между формулировками задачи HSLRA в векторной норме и матричной норме (3.8) и (3.9) были рассмотрены в [143] и [15]. Общий случай установлен в следующей теореме.

Теорема 6. Для любого $\mathbf{z} \in \mathbb{R}^N$, $\|\mathcal{T}_L(\mathbf{z})\|_{\mathbf{A}, \mathbf{B}} = \|\mathbf{z}\|_{\mathbf{W}}$ тогда и только тогда, когда

$$\mathbf{W} = \mathbf{A} * \mathbf{B}. \quad (3.10)$$

Доказательство. Рассмотрим квадрат нормы $\|\mathbf{X}\|_{\mathbf{A}, \mathbf{B}}^2 = \text{tr}(\mathbf{A} \mathbf{X} \mathbf{B} \mathbf{X}^T)$ с $\mathbf{X} = \mathcal{T}_L(\mathbf{z})$ так, что $x_{l,k} = z_{l+k}$ для $l = 0, \dots, L-1$ и $k = 0, \dots, K-1$. Мы имеем

$$\|\mathbf{X}\|_{\mathbf{A}, \mathbf{B}}^2 = \sum_{l, l', k, k'} a_{l, l'} x_{l', k'} b_{k', k} x_{l, k} = \sum_{l, l', k, k'} a_{l, l'} z_{l'+k'} b_{k', k} z_{l+k} = \sum_{l, l', k, k'} z_{n'} a_{l, l'} b_{n'-l', n-l} z_n,$$

где $n = k + l$, $n' = k' + l'$ и все суммы выше взяты для $l, l' = 0, \dots, L-1$ и $k, k' = 0, \dots, K-1$. Меняя индексы суммирования в последней сумме $k \rightarrow n$ и $k' \rightarrow n'$, получаем требуемое. \square

В типичном приложении, когда предполагается некоторая структура шума в модели ‘сигнал плюс шум’, задача HSLRA формулируется в векторной форме с заданной матрицей \mathbf{W} . Как уже упоминалось выше, алгоритмы решения задачи HSLRA намного проще, если мы имеем матричную, а не векторную форму задачи HSLRA. Поэтому, в свете теоремы 6, для заданной \mathbf{W} мы хотим найти положительно определенные матрицы \mathbf{A} и \mathbf{B} такие, что (3.10); то есть, мы хотим выполнить слепую деконволюцию матрицы \mathbf{W} .

Матрицы \mathbf{A} и \mathbf{B} в (3.9) и, следовательно, в (3.10) должны быть симметричными неотрицательно определенными, см. например, [144] и [30, стр.62]. В теоремах 2–4 ниже мы потребуем симметричности матриц \mathbf{A} и \mathbf{B} в (3.10).

Из результатов [13] следует, что в случае, когда шум ξ белый, и поэтому $\mathbf{W} = \mathbf{I}_N$, матрица \mathbf{W} не может быть слепо деконволютирована при условии, что \mathbf{A} и \mathbf{B} — положительно определенные матрицы; однако, для широкого диапазона параметров N и L существует множество пар неотрицательно определенных диагональных матриц \mathbf{A} и \mathbf{B} таких, что (3.10). Этот раздел расширяет результаты [13] на случай ленточных матриц, соответствующих случаю, когда шум ξ образует авторегрессионный процесс.

Модель белого шума является самой простой и, следовательно, самой популярной моделью шума, используемой для формулировки задач “сигнал плюс шум”. Авторегрессионная модель шума — вторая по распространённости модель шума, используемая в таких задачах. В частности, в климатологии наиболее распространенной моделью шума является так называемый ‘красный шум’, то есть авторегрессионный процесс порядка один с положительным коэффициентом. Поэтому, помимо понимания эквивалентности между векторной и матричной формами HSLRA, теоретическое исследование HSLRA, SSA и других методов, основанных на подпространстве, при наличии авторегрессионного шума также представляется важным.

3.2.3. Производящие функции и свертка ленточных матриц

Производящая функция матрицы через производящие функции ее диагоналей

В некоторых случаях (например, для ленточных матриц), естественно построить производящие функции матриц как сумму производящих функций диагоналей. Рассмотрим матрицу $\mathbf{U} = \{u_{i,j}\}_{i,j=0}^M$ и пусть $G_{\mathbf{U}}(t, s)$ будет ее производящей функцией.

Определим $\text{diag}_i(\mathbf{U})$, i -ю диагональ \mathbf{U} , как вектор длины $M - |i| + 1$ с элементами с индексами (j, k) , удовлетворяющими $k - j = i$, $i = -M, \dots, M$. Эта i -я диагональ $\text{diag}_i(\mathbf{U})$ имеет одномерную gf $u_i(\tau) = G_{\text{diag}_i(\mathbf{U})}(\tau) = \sum_{j=0}^{M-|i|} u_{j,j+i} \tau^j$.

Лемма 10.

$$G_{\mathbf{U}}(t, s) = \sum_{i=-M}^M t^{(|i|-i)/2} s^{(|i|+i)/2} u_i(ts).$$

Если \mathbf{U} симметрична, то эта формула упрощается до $G_{\mathbf{U}}(t, s) = u_0(ts) + \sum_{i=1}^M (t^i + s^i) u_i(ts)$.

Доказательство.

$$\begin{aligned} G_{\mathbf{U}}(t, s) &= \sum_{j,k=0}^M u_{j,k} t^j s^k = \sum_{i=-M}^M \sum_{k=j-i}^M u_{j,k} t^j s^k = \\ &= \sum_{i=1}^M t^i \sum_{j=0}^{M-i} u_{j+i,j} t^j s^j + \sum_{i=1}^M s^i \sum_{j=0}^{M-i} u_{j,j+i} t^j s^j + \sum_{i=0}^M u_{i,i} t^i s^i = \\ &= \sum_{i=0}^M t^i u_{-i}(ts) + \sum_{i=1}^M s^i u_i(ts) = \sum_{i=-M}^0 t^{-i} u_i(ts) + \sum_{i=1}^M s^i u_i(ts) = \\ &= \sum_{i=-M}^M t^{(|i|-i)/2} s^{(|i|+i)/2} u_i(ts). \end{aligned}$$

□

Свертка матриц, выраженная через свертку диагоналей; ленточные матрицы

Лемма 11. Пусть $\mathbf{A} = (a_{i,j})_{i,j=0}^A$, $\mathbf{B} = (b_{i,j})_{i,j=0}^B$ и $\mathbf{C} = \mathbf{A} * \mathbf{B}$. Для заданного целого числа i пусть $\mathbf{a}_i(t) = G_{\text{diag}_i(\mathbf{A})}(t) = \sum_{j=0}^{A-|i|} a_{j,j+i} t^j$ ($|i| \leq A$), $\mathbf{b}_i(t) = G_{\text{diag}_i(\mathbf{B})}(t)$ ($|i| \leq B$) и $\mathbf{c}_i(t) = G_{\text{diag}_i(\mathbf{C})}(t)$ ($|i| \leq C = A + B$) — производящие функции диагоналей $\text{diag}_i(\mathbf{A})$, $\text{diag}_i(\mathbf{B})$ и $\text{diag}_i(\mathbf{C})$ соответственно. Тогда

$$\mathbf{c}_i(t) = \sum_{j+k=i} t^{(|j|+|k|-|j+k|)/2} \mathbf{a}_j(t) \mathbf{b}_k(t) \quad (i = -C, \dots, C). \quad (3.11)$$

Доказательство. Ввиду леммы 10 мы должны доказать, что

$$\sum_{i=-C}^C t^{(|i|-i)/2} s^{(|i|+i)/2} \mathbf{c}_k(ts) = \sum_{i=-C}^C t^{(|i|-i)/2} s^{(|i|+i)/2} \sum_{j+k=i} (ts)^{(|j|+|k|-|j+k|)/2} \mathbf{a}_j(ts) \mathbf{b}_k(ts).$$

Мы имеем:

$$\begin{aligned} & \sum_{i=-C}^C t^{(|i|-i)/2} s^{(|i|+i)/2} \mathbf{c}_k(ts) \\ &= \left(\sum_{j=-A}^A t^{(|j|-j)/2} s^{(|j|+j)/2} \mathbf{a}_j(ts) \right) \left(\sum_{k=-B}^B t^{(|k|-k)/2} s^{(|k|+k)/2} \mathbf{b}_k(ts) \right) \\ &= \sum_{j=-A}^A \sum_{k=-B}^B t^{(|j|-j)/2} s^{(|j|+j)/2} \mathbf{a}_j(ts) t^{(|k|-k)/2} s^{(|k|+k)/2} \mathbf{b}_k(ts) \\ &= \sum_{i=-(A+B)}^{A+B} \sum_{j+k=i} t^{(|j|-j)/2} s^{(|j|+j)/2} t^{(|k|-k)/2} s^{(|k|+k)/2} \mathbf{a}_j(ts) \mathbf{b}_k(ts) \\ &= \sum_{i=-C}^C \sum_{j+k=i} t^{(|j|+|k|-|j+k|)/2} s^{(|j|+|k|+|j+k|)/2} \mathbf{a}_j(ts) \mathbf{b}_k(ts). \end{aligned}$$

Так как в сумме $j+k = i$, доказательство завершено. \square

Замечание 26. Для любых действительных j и k имеем:

$$(|j|+|k|-|j+k|)/2 = \begin{cases} 0, & \text{если } jk \geq 0 \\ |k| & \text{если } jk < 0, |j| \geq |k|; \\ |j| & \text{если } jk < 0, |j| < |k|. \end{cases}$$

Следующее следствие является переформулировкой леммы 11, используя явную форму для диагоналей \mathbf{C} .

Следствие 8. Пусть матрицы \mathbf{A} , \mathbf{B} и $\mathbf{C} = \mathbf{A} * \mathbf{B}$ такие, как в лемме 11. Тогда i -ая диагональ \mathbf{C} имеет вид

$$\text{diag}_i(\mathbf{C}) = \sum_{j+k=i} \begin{pmatrix} 0_{(|j|+|k|-|j+k|)/2} \\ \text{diag}_j(\mathbf{A}) * \text{diag}_k(\mathbf{B}) \\ 0_{(|j|+|k|-|j+k|)/2} \end{pmatrix}, \quad (3.12)$$

где $i = -C, \dots, C$, $C = A + B$, а $0_m \in \mathbb{R}^m$ — вектор из нулей размерности m .

В случае ленточных матриц следствие 8 принимает следующий вид.

Следствие 9. Пусть $\mathbf{A} = (a_{i,j})_{i,j=0}^A$ — $(2p_1+1)$ -диагональная матрица с $p_1 \leq A$, $\mathbf{B} = (b_{i,j})_{i,j=0}^B$ — $(2p_2+1)$ -диагональная с $p_2 \leq B$ и $\mathbf{C} = \mathbf{A} * \mathbf{B}$. Тогда \mathbf{C} является $(2p+1)$ -диагональю с $p = p_1 + p_2$ и i -диагональ $\text{diag}_i(\mathbf{C})$ матрицы \mathbf{C} ($i = -p, \dots, p$) задается (3.12); ее gf задается (3.11).

Когда $p_2 = 0$ и поэтому матрица \mathbf{B} диагональная, следствие 9 описывает следующий частный случай.

Следствие 10. Пусть \mathbf{A} — $(2p+1)$ -диагональная матрица, а \mathbf{B} — диагональная с \mathbf{b} на главной диагонали. Тогда матрица $\mathbf{C} = \mathbf{A} * \mathbf{B}$ является $(2p+1)$ -диагональной с $\text{diag}_i(\mathbf{C}) = \text{diag}_i(\mathbf{A}) * \mathbf{b}$, $i = -p, \dots, p$; в терминах gf мы имеем $c_i(t) = a_i(t)b_0(t)$, $i = -p, \dots, p$.

3.2.4. Автоковариационные матрицы и их инверсии для AR(1) и AR(2) моделей

Обратные автоковариационные матрицы

Хорошо известно, что обратная автоковариационная матрица $\text{AR}(p)$ является положительно-определенной симметричной $(2p+1)$ -диагональной матрицей [145, p.534]. Ниже мы рассмотрим явные формы Σ^{-1} для моделей AR(1) и AR(2).

AR(1) Пусть $\xi = (\xi_0, \xi_1, \dots, \xi_W)^T$ является AR(1) процессом с

$$\xi_j = \phi_1 \xi_{j-1} + \varepsilon_j, \quad (3.13)$$

где $j = 1, \dots, W$, $\phi_1 \neq 0$, и $\varepsilon_1, \varepsilon_2, \dots$ являются независимыми одинаково распределенными случайными величинами с распределением $\mathcal{N}(0, 1)$; ξ_j имеют распределение $\mathcal{N}(0, \sigma^2)$ для всех $j = 0, \dots, W$, где $\sigma^2 = 1/(1 - \phi_1^2)$. Условием стационарности процесса (3.13) является $|\phi_1| < 1$. Ковариационная матрица Σ вектора ξ имеет вид $\Sigma = \sigma^2 (\phi_1^{|i-j|})_{i,j=0}^W$. Обратной к Σ является трехдиагональная матрица $\Sigma^{-1} = \mathbf{W} \in \mathbb{R}^{(W+1) \times (W+1)}$, где

$$\mathbf{W} = \begin{pmatrix} 1 & k_1 & 0 & 0 & \dots & & \\ k_1 & k_0 & k_1 & 0 & \dots & & \\ 0 & k_1 & k_0 & k_1 & 0 & & \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & 0 & k_1 & k_0 & k_1 & \\ & & 0 & 0 & k_1 & 1 & \end{pmatrix}, \quad (3.14)$$

$k_0 = 1 + \phi_1^2$, $k_1 = -\phi_1$. Заметим, что матрица (3.14) определена для любого ϕ_1 , необязательно для $|\phi_1| < 1$.

AR(2) Пусть $\xi = (\xi_0, \xi_1, \dots, \xi_W)^T$ является AR(2) процессом с

$$\xi_j = \phi_1 \xi_{j-1} + \phi_2 \xi_{j-2} + \varepsilon_j \quad (j = 2, \dots, W) \quad (3.15)$$

где ε_j являются независимыми одинаково распределенными случайными величинами с распределением $\mathcal{N}(0, 1)$; ξ_j имеют распределение $\mathcal{N}(0, \sigma^2)$ для всех $j = 0, \dots, W$; здесь $\sigma^2 = (1 - \phi_2)/[(1 + \phi_1 - \phi_2)(1 - \phi_1 - \phi_2)(1 + \phi_2)]$. Условиями стационарности являются $\phi_2 + |\phi_1| < 1$ и $|\phi_2| < 1$. Область стационарности модели AR(2) изображена на рис. 3.3. Обратной к ковариационной матрицей Σ является пятидиагональная матрица $\Sigma^{-1} = \mathbf{W} = (w_{i,j})_{i,j=0}^W$, где

$$\mathbf{W} = \begin{pmatrix} 1 & k_{12} & k_2 & 0 & 0 & 0 & 0 & 0 & \dots \\ k_{21} & k_{22} & k_1 & k_2 & 0 & 0 & 0 & \dots \\ k_2 & k_1 & k_0 & k_1 & k_2 & 0 & 0 & \dots \\ 0 & k_2 & k_1 & k_0 & k_1 & k_2 & 0 & \dots \\ \vdots & \ddots \\ & & & 0 & k_2 & k_1 & k_0 & k_1 & k_2 \\ & & & 0 & 0 & k_2 & k_1 & k_{22} & k_{12} \\ & & & 0 & 0 & 0 & 0 & k_2 & k_{21} & 1 \end{pmatrix}, \quad (3.16)$$

$k_0 = 1 + \phi_1^2 + \phi_2^2$, $k_1 = -\phi_1 + \phi_1 \phi_2$, $k_2 = -\phi_2$, $k_{12} = k_{21} = -\phi_1$, $k_{22} = 1 + \phi_1^2$. Матрица (3.16) определена для любых ϕ_1 и ϕ_2 .

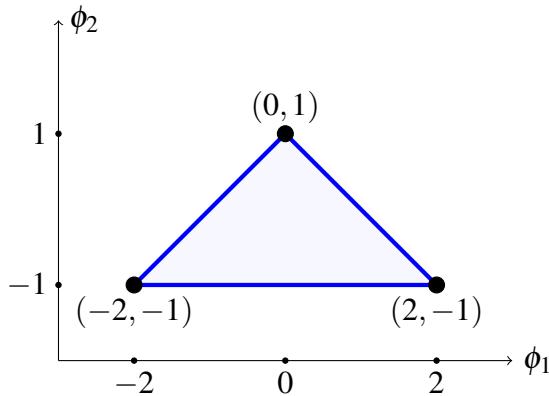


Рис. 3.3: Область стационарности модели AR(2) и критические точки

Инверсии автоковариационных матриц и их производящие функции

Пусть $\mathbf{W} = c\Sigma^{-1} = (w_{i,j})_{i,j=0}^W$, где Σ — автоковариационная матрица процесса AR(p) с некоторым $p > 0$, а c выбирается так, чтобы $w_{0,0} = 1$, как в (3.14) и (3.16). Напомним, что матрица \mathbf{W} является ленточной.

Рассмотрим связь между диагоналями \mathbf{W} и их производящими функциями. Обозначим $w_k(t) \stackrel{\text{def}}{=} G_{\text{diag}_k(\mathbf{W})}(t)$, $t \in \mathbb{C}$. Для $k \geq 0$ явная формула для w_k имеет вид $w_k(t) = \sum_{j=0}^{W-k} w_{j,j+k} t^j$.

Производящая функция вектора из единиц Ниже мы будем часто использовать следующую gf. Пусть $\mathbf{c}_M = (1, \dots, 1)^T \in \mathbb{R}^{M+1}$. Тогда

$$C_M(t) \stackrel{\text{def}}{=} G_{\mathbf{c}_M}(t) = 1 + \dots + t^M = \frac{1 - t^{M+1}}{1 - t}.$$

Следующая удобная формула связывает многочлены $C_{M+k}(t)$ и $C_M(t)$ при $k > 0$:

$$C_{M+k}(t) = t^k C_M(t) + C_{k-1}(t). \quad (3.17)$$

AR(1) Рассмотрим модель AR(1) (3.13) с коэффициентом регрессии ϕ_1 . Из (3.14) следует, что

$$w_1(t) = -\phi_1 C_{W-1}(t) \quad w_0(t) = C_W(t) + \phi_1^2 t C_{W-2}(t).$$

Применяя (3.17) с $(k, M) = (1, W-1)$ и $(k, M) = (1, W-2)$, получаем следующую лемму.

Лемма 12. Для модели AR(1) (3.13),

$$w_1(t) = -\phi_1 C_{W-1}(t), \quad w_0(t) = t C_{W-1}(t) + 1 + \phi_1^2 (C_{W-1}(t) - 1). \quad (3.18)$$

AR(2) Рассмотрим модель AR(2)(3.15) с коэффициентами регрессии ϕ_1 и ϕ_2 . Из (3.16) получаем

$$\begin{aligned} w_2(t) &= -\phi_2 C_{W-2}(t), \\ w_1(t) &= -\phi_1 C_{W-1}(t) + \phi_1 \phi_2 t C_{W-3}(t), \\ w_0(t) &= C_W(t) + \phi_1^2 t C_{W-2}(t) + \phi_2^2 t^2 C_{W-4}(t). \end{aligned}$$

Применяя (3.17) с $k = 1, M = W-3, W-2$ и с $k = 2, M = W-4, W-2$, получаем следующую лемму.

Лемма 13. Для модели AR(2) (3.15),

$$\begin{aligned} w_2(t) &= -\phi_2 C_{W-2}(t), \\ w_1(t) &= -\phi_1 (t C_{W-2}(t) + 1) + \phi_1 \phi_2 (C_{W-2}(t) - 1), \\ w_0(t) &= t^2 C_{W-2}(t) + t + 1 + \phi_1^2 t C_{W-2}(t) + \phi_2^2 (C_{W-2}(t) - (t + 1)). \end{aligned} \quad (3.19)$$

3.2.5. Изучение существования решений задачи слепой деконволюции для матриц, пропорциональных обратным ковариационным матрицам в авторегрессионных моделях

Для заданных квадратных матриц \mathbf{A} и \mathbf{B} обозначим $a_j(t) = G_{\text{diag}_j(\mathbf{A})}(t)$ и $b_j(t) = G_{\text{diag}_j(\mathbf{B})}(t)$.

AR(1)

Теорема 7. Пусть $W \geq 2$ и матрица \mathbf{W} определяется (3.14). Существуют симметричные матрицы $\mathbf{A} = (a_{i,j})_{i,j=0}^A$ и $\mathbf{B} = (b_{i,j})_{i,j=0}^B$ при $A, B \geq 1$ такие, что $\mathbf{W} = \mathbf{A} * \mathbf{B}$, если и только если $|\phi_1| = 1$.

Доказательство. Предположим, что $\mathbf{W} = \mathbf{A} * \mathbf{B}$. Так как \mathbf{W} 3-диагональна, то в силу следствия 10 и предположения о симметрии \mathbf{A} и \mathbf{B} , мы можем рассмотреть только случай, когда \mathbf{A} 3-диагональна, а \mathbf{B} диагональна.

Из (3.11) получаем

$$w_1(t) = a_1(t)b_0(t), \quad w_0(t) = a_0(t)b_0(t), \quad \forall t \in \mathbb{C}. \quad (3.20)$$

Так как $\phi_1 \neq 0$, из левых равенств в (3.18) и (3.20) вытекает, что все корни многочлена $C_{W-1}(t)$ являются корнями многочлена $a_1(t)b_0(t)$. Так как $B \geq 1$, то хотя бы один из корней многочлена $C_{W-1}(t)$ является корнем многочлена $b_0(t)$. Пусть $t_1 \in \mathbb{C}$ — такой корень.

Предположим, что $|\phi_1| \neq 1$. Из второго равенства в (3.20) следует $w_0(t_1) = 0$. но второе равенство в (3.18) дает $w_0(t_1) = 1 - \phi_1^2 \neq 0$. Это противоречие доказывает необходимость $|\phi_1| = 1$.

Предположим теперь, что $|\phi_1| = 1$ так, что $\phi_1 = \pm 1$. В этом случае $w_0(t) = (t+1)C_{W-1}(t)$ и поэтому, принимая также во внимание первое равенство в (3.18), равенства (3.20) становятся такими

$$a_1(t)b_0(t) = -\phi_1 C_{W-1}(t), \quad a_0(t)b_0(t) = (t+1)C_{W-1}(t). \quad (3.21)$$

Представим $C_{W-1}(t)$, который является многочленом степени $W-1 \geq 1$, как произведение двух ненулевых многочленов $p(t)$ и $q(t)$:

$$C_{W-1}(t) = p(t)q(t). \quad (3.22)$$

Здесь многочлен $p(t)$ может быть константой, а $q(t)$ имеет степень не ниже 1. Тогда мы можем выбрать

$$b_0(t) = q(t), \quad a_0(t) = (t+1)p(t) \quad a_1(t) = -\phi_1 p(t). \quad (3.23)$$

□

AR(2)

Рассмотрим модель AR(2) (3.15) с коэффициентами регрессии ϕ_1 и ϕ_2 . Мы стремимся определить матрицы \mathbf{A} и \mathbf{B} так, чтобы (3.10) выполнялась; то есть, $\mathbf{W} = \mathbf{A} * \mathbf{B}$. Так как \mathbf{W} пятидиагональна и в силу следствия 9, это может произойти только в следующих двух

случаях: (a) каждая из \mathbf{A} и \mathbf{B} трехдиагональна и (b) \mathbf{A} — 5-диагональна, а \mathbf{B} — диагональна.

А и В симметричны и обе матрицы 3-диагональные

Теорема 8. Пусть $W \geq 3$ и матрица \mathbf{W} определена в (3.16) с $\phi_2 \neq 0$. Предположим, что матрицы $\mathbf{A} = (a_{i,j})_{i,j=0}^A$ и $\mathbf{B} = (b_{i,j})_{i,j=0}^B$ являются 3-диагональными и $A \geq 2$, $B \geq 1$. Существуют такие матрицы \mathbf{A} и \mathbf{B} с $\mathbf{W} = \mathbf{A} * \mathbf{B}$, если и только если (i) $\phi_2 = -1$ и $|\phi_1| \geq 2$ или (ii) $\phi_2 = 1$ и $\phi_1 = 0$.

Доказательство теоремы 8 основано на нескольких леммах.

Лемма 14. В условиях теоремы 8, существование матриц \mathbf{A} и \mathbf{B} таких, что $\mathbf{W} = \mathbf{A} * \mathbf{B}$, подразумевает, что многочлен $w_1^2(t) - 4w_2(t)(w_0(t) - 2tw_2(t))$ является квадратом многочлена от t .

Доказательство. Из (3.11) получаем

$$\begin{aligned} w_2(t) &= a_1(t)b_1(t), \\ w_1(t) &= a_0(t)b_1(t) + a_1(t)b_0(t), \\ w_0(t) &= a_0(t)b_0(t) + 2ta_1(t)b_1(t). \end{aligned}$$

Для краткости мы опустим аргумент t . Мы имеем:

$$a_1b_1 = w_2, \quad a_0b_1 + a_1b_0 = w_1, \quad a_0b_0 = w_0 - 2tw_2.$$

Обозначим $y = a_0b_1$, $z = a_1b_0$. Тогда получим

$$y + z = w_1, \quad yz = w_2(w_0 - 2tw_2).$$

Это означает, что y и z являются корнями квадратного уравнения $x^2 + w_1x + w_2(w_0 - 2tw_2) = 0$. Эти корни

$$x_{1,2} = \frac{w_1 \pm \sqrt{w_1^2 - 4w_2(w_0 - 2tw_2)}}{2}. \quad (3.24)$$

Поскольку корни должны быть многочленами, $w_1^2 - 4w_2(w_0 - 2tw_2)$ должно быть квадратом многочлена от t (напомним, что w_i — многочлены от t). \square

Лемма 15. Предположим, что многочлен $P_{2M}(t)$ имеет вид

$$P_{2M}(t) = (P_1(t)P_{M-1}(t))^2 + Q_1(t)P_{M-1}(t) + P_0$$

для некоторого $M \geq 1$ (нижние индексы означают степени многочлена). Если квадратный корень из $P_{2M}(t)$ существует, то он имеет вид $\pm P_1(t)P_{M-1}(t) + \text{const}$.

Доказательство. Пусть $P_1(t) = at + b$, $Q_1(t) = dt + f$. Тогда для $\gamma = d/(2a)$ имеем

$$P_{2M}(t) = (P_1(t)P_{M-1}(t) + \gamma)^2 + gP_{M-1}(t) + p_0,$$

где g и p_0 — некоторые константы. Пусть $P_{2M}(t) = U_M^2(t)$. Обозначим $V_M(t) = P_1(t)P_{M-1}(t) + \gamma$, $D_{M-1}(t) = gP_{M-1}(t) + p_0$. Тогда

$$D_{M-1}(t) = U_M^2(t) - V_M^2(t) = (U_M(t) - V_M(t))(U_M(t) + V_M(t)).$$

Левая часть этого уравнения является многочленом порядка $M - 1$ или меньше, но правая часть либо тождественный ноль, либо многочлен порядка M или больше. Поэтому это равенство может быть справедливо только в том случае, если $U_M(t) = \pm V_M(t)$. \square

Лемма 16. Для матрицы \mathbf{W} , определенной в (3.16) с $\phi_2 \neq 0$, многочлен $w_1^2(t) - 4w_2(t)(w_0(t) - 2tw_2(t))$ является квадратным многочленом тогда и только тогда, когда выполняется хотя бы одно из следующих соотношений:

(A) $\phi_2 = -1$, (B) $\phi_2 = 1 - \phi_1$, (C) $\phi_2 = 1 + \phi_1$.

Доказательство. Для краткости мы опустим аргумент многочлена t . Обозначим $C = C_{W-2}(t)$. Прямая подстановка с помощью (3.19) дает

$$\begin{aligned} w_1^2 - 4(w_0 - 2tw_2)w_2 &= (-\phi_1 t + \phi_1 \phi_2)C - \phi_1(1 + \phi_2))^2 + \\ &+ 4\phi_2(t^2 + (\phi_1^2 + 2\phi_2)t + \phi_2^2)C^2 + 4C\phi_2(t + 1 - \phi_2^2(t + 1)). \end{aligned}$$

Рассмотрим этот многочлен как многочлен от C . Через v_i обозначим коэффициент для C^i , $i = 0, 1, 2$. Мы имеем:

$$\begin{aligned} v_0 &= \phi_1^2(1 + \phi_2)^2, \\ v_1 &= 2(1 + \phi_2)(-\phi_1^2(\phi_2 - t) + 2\phi_2(t + 1)(1 - \phi_2)), \\ v_2 &= (\phi_1^2 + 4\phi_2)(t + \phi_2)^2. \end{aligned}$$

Ввиду леммы 15, определитель

$$v_1 - 4v_0v_2 = 4(1 + \phi_2)^2\phi_2(1 + \phi_1 - \phi_2)(1 - \phi_1 - \phi_2)((t + 1)^2\phi_2 + t\phi_1^2)$$

должен быть равен нулю. Решениями уравнения $v_1 - 4v_0v_2 = 0$ относительно ϕ_2 являются: $\phi_2 = 0, -1, 1 - \phi_1, 1 + \phi_1, -t\phi_1^2/(1+t)^2$. Корень $\phi_2 = 0$ не подходит, так как $\phi_2 \neq 0$ по определению AR(2). Корень $\phi_2 = -t\phi_1^2/(1+t)^2$ не подходит, так как зависит от t . На этом доказательство завершено. \square

Доказательство теоремы 8. .

Ввиду лемм 14 и 16, матрицы \mathbf{A} и \mathbf{B} такие, что $\mathbf{W} = \mathbf{A} * \mathbf{B}$ существуют только в трех частных случаях, указанных в лемме 16; а именно, (A) $\phi_2 = -1$, (B) $\phi_2 = 1 - \phi_1$, (C) $\phi_2 = 1 + \phi_1$. Рассмотрим эти три случая по отдельности.

(A) Предположим, что $\phi_2 = -1$. Тогда

$$w_2(t) = C_{W-2}(t), \quad w_1(t) = -\phi_1 C_{W-2}(t)(t+1), \quad w_0(t) = C_{W-2}(t)(t^2 + \phi_1^2 t + 1).$$

Из (3.19) и (3.24) мы имеем:

$$\begin{aligned} a_1(t)b_1(t) &= C_{W-2}(t), \\ a_0(t)b_1(t) + a_1(t)b_0(t) &= -\phi_1 C_{W-2}(t)(t+1), \\ a_0(t)b_0(t) + 2ta_1(t)b_1(t) &= C_{W-2}(t)(t^2 + \phi_1^2 t + 1). \end{aligned} \tag{3.25}$$

Так как многочлены $a_0(t)$ и $b_0(t)$ имеют на одну степень выше, чем многочлены $a_1(t)$ и $b_1(t)$ соответственно, $a_1(t)b_1(t) = C_{W-2}(t)$ и $a_0(t)b_0(t)$ содержит $C_{W-2}(t)$ в качестве множителя, получаем, что $a_0(t) = (\lambda_0 + \lambda_1 t)a_1(t)$ и $b_0(t) = (\mu_0 + \mu_1 t)b_1(t)$ для некоторых $\lambda_0, \lambda_1, \mu_0$ и μ_1 . Подставляя это в уравнения (3.25) и убирая $C_{W-2}(t)$, который является общим множителем во всех уравнениях, мы получаем следующие два уравнения для $\lambda_0, \lambda_1, \mu_0$ и μ_1 :

$$\begin{cases} (\mu_0 + \mu_1 t)(\lambda_0 + \lambda_1 t) &= t^2 - 2t + \phi_1^2 t + 1, \\ (\mu_0 + \mu_1 t) + (\lambda_0 + \lambda_1 t) &= -(t+1)\phi_1. \end{cases}$$

Приравнивая коэффициенты двух многочленов от t , находим, что не существует решений для $\lambda_0, \lambda_1, \mu_0$ и μ_1 , когда $|\phi_1| < 2$. С другой стороны, существуют следующие решения, когда $|\phi_1| \geq 2$: пусть

$$z_1 = \frac{-\phi_1 + \sqrt{\phi_1^2 - 4}}{2} \quad z_2 = \frac{-\phi_1 - \sqrt{\phi_1^2 - 4}}{2} \tag{3.26}$$

— два решения уравнения $z^2 + \phi_1 z + 1 = 0$, тогда мы можем выбрать либо $\mu_0 = \lambda_1 = z_1$, $\lambda_0 = \mu_1 = z_2$ либо $\mu_0 = \lambda_1 = z_2$, $\lambda_0 = \mu_1 = z_1$. Это дает необходимые выражения для $a_0(t)$, $a_1(t)$, $b_0(t)$ и $b_1(t)$.

(B) Предположим, что $\phi_2 = 1 - \phi_1$. Тогда уравнения (3.19) можно переписать как

$$\begin{aligned} a_1(t)b_1(t) &= -(1 - \phi_1)C_{W-2}(t), \\ a_0(t)b_1(t) + a_1(t)b_0(t) &= -\phi_1(tC_{W-2}(t) + 1) + \phi_1(1 - \phi_1)(C_{W-2}(t) - 1), \\ a_0(t)b_0(t) + 2ta_1(t)b_1(t) &= t^2C_{W-2}(t) + t + 1 + \phi_1^2 t C_{W-2}(t) + (1 - \phi_1)^2(C_{W-2}(t) - (t+1)). \end{aligned} \tag{3.27}$$

Используя выражение (3.24) для произведений $a_0(t)b_1(t)$ и $a_1(t)b_0(t)$, получаем $a_1(t)b_0(t) = (1 - \phi_1)(t+1)C_{W-2}(t)$ как один из двух корней, заданных (3.24). Как и в случае (A), из первого уравнения в (3.27) получаем $b_0(t) = -(1+t)b_1(t)$. Возьмем любой корень t_1 многочлена $a_1(t)$ и

подставим его во второе уравнение в (3.27). Поскольку корни $a_1(t)$ состоят из корней $C_{W-2}(t)$, получаем $\phi_1(2 - \phi_1) = 0$. Решение $\phi_1 = 0$ подразумевает $\phi_2 = 1$, а $\phi_1 = 2$ дает $\phi_2 = -1$. Случай $\phi_1 = 2, \phi_2 = -1$ дает решение, описанное выше в случае (A). В случае $\phi_1 = 0, \phi_2 = 1$ мы имеем решения (3.24), полученные из произвольного расщепления

$$C_{W-2}(t) = p(t)q(t) \quad (3.28)$$

многочлена $C_{W-2}(t)$ в произведение двух ненулевых многочленов $p(t)$ и $q(t)$, задав

$$a_0(t) = (1+t)p(t), \quad a_1(t) = p(t), \quad b_0(t) = (1+t)q(t) \text{ и } b_1(t) = -q(t). \quad (3.29)$$

(C) Предположим, что $\phi_2 = 1 + \phi_1$. Уравнения (3.19) становятся равными

$$\begin{aligned} a_1(t)b_1(t) &= -(1 + \phi_1)C_{W-2}(t), \\ a_0(t)b_1(t) + a_1(t)b_0(t) &= -\phi_1(tC_{W-2}(t) + 1) + \phi_1(1 + \phi_1)(C_{W-2}(t) - 1), \\ a_0(t)b_0(t) + 2ta_1(t)b_1(t) &= t^2C_{W-2}(t) + t + 1 + \phi_1^2tC_{W-2}(t) + \\ &\quad (1 + \phi_1)^2(C_{W-2}(t) - (t + 1)). \end{aligned}$$

Одним из двух решений для $a_1(t)b_0(t)$ является $a_1(t)b_0(t) = C_{W-2}(t + (\phi_1 + 1)^2)$. Аналогично вышесказанному получаем $\phi_1(2 + \phi_1) = 0$. Решение $\phi_1 = 0$ дает $\phi_2 = 1$ и, следовательно, тот же набор решений уравнений (3.19), что и в случае (B). Решение $\phi_1 = -2$ дает $\phi_2 = -1$ и рассматривается в примере (A). \square

A симметрична и 5-диагональна, B диагональна

Теорема 9. Пусть $W \geq 4$ и матрица \mathbf{W} определена в (3.16). Существуют матрицы $\mathbf{A} = (a_{i,j})_{i,j=0}^A$ и $\mathbf{B} = (b_{i,j})_{i,j=0}^B$ с $A \geq 2$ и $B \geq 2$, где \mathbf{A} 5-диагональная, а \mathbf{B} диагональная, что $\mathbf{W} = \mathbf{A} * \mathbf{B}$ тогда и только тогда, когда (a) $\phi_2 = -1$ или (b) $\phi_1 = 0$ и $\phi_2 = 1$.

Доказательство. Предположим, что $\mathbf{W} = \mathbf{A} * \mathbf{B}$. Из (3.11) получаем

$$w_2(t) = a_2(t)b_0(t), \quad w_1(t) = a_1(t)b_0(t), \quad w_0(t) = a_0(t)b_0(t), \quad \forall t \in \mathbb{C}. \quad (3.30)$$

Поскольку $\phi_2 \neq 0$, из левых равенств в (3.19) и (3.30) вытекает, что все корни многочлена $C_{W-2}(t)$ являются корнями многочлена $a_2(t)b_0(t)$. Так как по предположению $B \geq 2$, то по крайней мере два корня многочлена $C_{W-2}(t)$ являются корнями многочлена $b_0(t)$. Пусть t_1 и t_2 — два таких корня.

Предположим, что условия на ϕ_1 и ϕ_2 не выполняются. Из второго равенства в (3.19) следует $w_1(t_i) = 0$, но второе равенство в (3.30) дает $w_1(t_i) = -\phi_1(1 + \phi_2)$, $i = 1, 2$. Из третьего равенства в (3.19) следует $w_0(t_i) = 0$, но третье равенство в (3.30) дает $w_0(t_i) = (1 - \phi_2^2)(t_i + 1)$,

$i = 1, 2$. Это возможно только в случаях (а) $\phi_2 = -1$ и (б) $\phi_1 = 0, \phi_2 = 1$. Это противоречие доказывает необходимость условий на ϕ_1 и ϕ_2 .

Предположим (а): $\phi_2 = -1$. Тогда

$$w_2(t) = C_{W-2}(t), w_1(t) = -\phi_1(C_{W-2}(t)(t+1)), w_0(t) = C_{W-2}(t)(t^2 + \phi_1^2 t + 1).$$

Следовательно,

$$\begin{aligned} a_0(t)b_0(t) &= C_{W-2}(t)(t^2 + \phi_1^2 t + 1), \\ a_1(t)b_0(t) &= -\phi_1(C_{W-2}(t)(t+1)), \\ a_2(t)b_0(t) &= C_{W-2}(t). \end{aligned}$$

Пусть $C_{W-2}(t) = p(t)q(t)$, где $p(t)$ — ненулевой многочлен любой степени, включая 0, а $q(t)$ — многочлен степени не ниже 2. Тогда мы можем выбрать

$$b_0(t) = q(t), a_2(t) = p(t), a_1(t) = -\phi_1(t+1)p(t) \text{ и } a_0(t) = (t^2 + \phi_1^2 t + 1)p(t). \quad (3.31)$$

Теперь предположим (б): $\phi_1 = 0, \phi_2 = 1$. Тогда

$$w_2(t) = -C_{W-2}(t), w_1(t) = 0, w_0(t) = (t^2 + 1)C_{W-2}(t).$$

Следовательно,

$$a_0(t)b_0(t) = (t^2 + 1)C_{W-2}(t), a_1(t)b_0(t) = 0, a_2(t)b_0(t) = -C_{W-2}(t).$$

Опять же, пусть $C_{W-2}(t) = p(t)q(t)$ с теми же предположениями о $p(t)$ и $q(t)$. Тогда мы можем выбрать $b_0(t) = q(t), a_2(t) = -p(t), a_1(t) = 0$ и $a_0(t) = (t^2 + 1)p(t)$. \square

Замечание 27. По технической причине теорема 9 не охватывает случай $B = 1$. Этот случай может быть рассмотрен отдельно следующим образом. Во-первых, аналогично обсуждению в разделе 3.2.5, многочлен $C_{W-2}(t)$ делится на многочлен $q(t) = b_0(t)$ степени 1 с вещественными коэффициентами тогда и только тогда, когда W является нечетным; этот многочлен $b_0(t) = \lambda(1+t)$ с $\lambda \neq 0$. Рассмотрим для нечетного W равенства (3.19) и (3.30). Первое и третье равенства в (3.19) показывают, что $a_0(t) = w_0(t)/b_0(t)$ и $a_2(t) = w_2(t)/b_0(t)$ являются многочленами для произвольных значений ϕ_1 и ϕ_2 . Однако, второе равенство в (3.19) подразумевает, что $a_1(t) = w_1(t)/b_0(t)$ является многочленом тогда и только тогда, когда $w_1(-1) = 0$, т.е. либо $\phi_2 = -1$, либо $\phi_1 = 0$ (в последнем случае $a_1(t) = 0$).

Замечание 28. Основными шагами в доказательствах теорем 7 и 9 является вычисление остатков от деления $w_k(t)$ на $C_{W-p}(t)$, где $p = 1$ и $k = 0$ для $AR(1)$ и $p = 2$ и $k = 0, 1$ для $AR(2)$. Действительно, обозначим такой остаток от деления как $r_k(t)$ и B корней многочлена $b_0(t)$, которые также являются корнями многочлена $C_{W-p}(t)$, как t_1, \dots, t_p (эти

корни существуют, так как $B \geq p$). Тогда $w_k(t_i) = r_k(t_i)$ для любых $i = 1, \dots, p$. Если p больше, чем степени остатков $r_k(t)$ для каждого k , то $\{t_i\}_{i=1}^p$ являются корнями $w_k(t)$ тогда и только тогда, когда остатки $r_k(t_i)$ равны нулю. Это случай теоремы 7 и теоремы 9.

Замечание 29. Теоремы 7, 8 и 9 показывают, что для стационарных моделей AR(1) и AR(2) деконволюция $\mathbf{W} = \mathbf{A} * \mathbf{B}$ при $A, B \geq 2$ не может быть выполнена, так как необходимые и достаточные условия в этих теоремах противоречат условиям стационарности.

3.2.6. Несуществование деконволюции $\mathbf{W} = \mathbf{A} * \mathbf{B}$ для стационарной AR(p) модели с общим p и диагональю \mathbf{B}

Рассмотрим модель AR(p) в форме $\sum_{i=0}^p \alpha_i \xi_{n-i} = \varepsilon_n$, где $D\varepsilon_n = 1$ и $\alpha_p \neq 0$. Это выражение связано с основным выражением в виде $\xi_n = \sum_{i=1}^p \phi_i \xi_{n-i} + \varepsilon_n$ простыми соотношениями $\alpha_0 = 1$ и $\alpha_i = -\phi_i$ для $i > 0$.

Пусть Σ — автоковариационная матрица этого AR(p) процесса. В соответствии с [146] и [145, Eq. 10], матрица $\mathbf{W} = \Sigma^{-1}$ имеет вид симметричной $(2p+1)$ -диагональной матрицы с

$$w_{i,j} = \begin{cases} \sum_{m=0}^{\min(i,j)} \alpha_m \alpha_{m+|i-j|} & \text{для } \max(i,j) < p; \\ \sum_{m=0}^{p-|i-j|} \alpha_m \alpha_{m+|i-j|} & \text{для } \max(i,j) \geq p, \min(i,j) \leq W-p, |i-j| = 0, \dots, p; \\ \sum_{m=0}^{\min(W-i, W-j)} \alpha_m \alpha_{m+|i-j|} & \text{для } \min(i,j) > W-p; \\ 0, & \text{иначе,} \end{cases}$$

где $i, j = 0, \dots, W$.

Одним из условий стационарности для модели AR(p) является условие $|\phi_p| < 1$, см. например, тест Jury на стабильность [147, Раздел 3.9]. В этом разделе мы покажем, что если $|\phi_p| < 1$, то деконволюция (3.10) невозможна при предположении, что матрица \mathbf{B} диагональная и имеет размер $p \times p$.

Таким же образом, как и раньше, мы можем выразить диагонали \mathbf{W} через производящие функции $C_M(t)$ векторов $(1, \dots, 1)^T \in \mathbb{R}^{M+1}$ при $M = W-2p, W-2p+1, \dots, W$:

$$w_k(t) = \sum_{i=0}^{p-k} \alpha_i \alpha_{i+k} t^i C_{W-2i-k}, \quad k = 0, \dots, p. \quad (3.32)$$

Например, $w_p(t) = -\phi_p C_{W-p}(t)$.

Выражения для $w_k(t)$ в терминах $C_{W-p}(t)$ играют важную роль для получения результатов для AR(1) и AR(2). В частности, теоремы 7 и 9, где \mathbf{A} или \mathbf{B} в деконволюции $\mathbf{W} = \mathbf{A} * \mathbf{B}$ является диагональным, основаны на рассмотрении остатков от деления $w_k(t)$ на $C_{W-p}(t)$ для $k = 0, 1$ и AR(1) и $k = 0, 1, 2$ и AR(2), см. замечание 28.

Из (3.32) следует, что $w_p(t)$ пропорционально $C_{W-p}(t)$ для любого p . Поэтому, аналогично доказательствам теорем 7 и 9, можно доказать, что необходимое условие для возможности деконволюции $\mathbf{W} = \mathbf{A} * \mathbf{B}$, где \mathbf{B} — диагональная матрица и $B > p$, — это нулевой остаток

от деления $w_k(t)$ на $C_{W-p}(t)$ для $k = 0, \dots, p-1$. В доказательстве используются равенства $w_k(t) = a_k(t)b_0(t)$, $k \leq p$, полученные в следствии 10.

Следующая лемма вычисляет остатки.

Лемма 17. *Остаток $r_k(t)$ от деления $w_k(t)$ на $C_{W-p}(t)$ для $k = 0, \dots, p$ равен*

$$r_k(t) = \sum_{i=0}^{p-k} \alpha_i \alpha_{i+k} (C_{p-k-i-1}(t) - C_{i-1}(t)). \quad (3.33)$$

(Здесь мы предполагаем, что $C_{-1} = 0$).

Доказательство. Непосредственно следует из (3.32) и равенств $t^i C_{W-2i-k} = C_{W-i-k} - C_{i-1}$ и $C_{W-i-k} = t^{p-(i+k)} C_{W-p} + C_{p-k-i-1}$, которые являются частными случаями (3.17). \square

Следствие 11.

$$r_p(t) = 0, \quad (3.34)$$

$$r_{p-1}(t) = \alpha_0 \alpha_{p-1} - \alpha_1 \alpha_p, \quad (3.35)$$

$$r_0(t) = \sum_{m=0}^{p-1} \left(\sum_{i=0}^{\min(m, p-m-1)} (\alpha_i^2 - \alpha_{p-i}^2) \right) t^m. \quad (3.36)$$

Доказательство. Равенство (3.34) легко получается из (3.33), так как $C_{-1} = 0$. Равенство (3.35) доказывается прямой заменой $k = p-1$ и $i = 0, 1$ в (3.33).

Для доказательства (3.36) подставим $k = 0$ в сумму (3.33):

$$r_0(t) = \sum_{i=0}^p \alpha_i^2 (C_{p-i-1}(t) - C_{i-1}(t)) \quad (3.37)$$

и затем запишем эту сумму в виде многочлена от t . По (3.17) получаем для $i = 0, \dots, p$

$$C_{p-i-1}(t) - C_{i-1}(t) = \begin{cases} t^i C_{p-2i-1}, & i < p/2, \\ 0, & i = p/2, p \text{ четное}, \\ -t^{p-i} C_{2i-p-1}, & i > p/2. \end{cases}$$

Следовательно,

$$r_0(t) = \sum_{i=0}^{\lfloor (p-1)/2 \rfloor} \alpha_i^2 t^i C_{p-2i-1} - \sum_{j=\lceil (p+1)/2 \rceil}^p \alpha_j^2 t^{p-j} C_{2j-p-1}.$$

Беря $j = p-i$, получаем

$$r_0(t) = \sum_{i=0}^{\lfloor (p-1)/2 \rfloor} (\alpha_i^2 - \alpha_{p-i}^2) t^i C_{p-2i-1} = \sum_{i=0}^{\lfloor (p-1)/2 \rfloor} (\alpha_i^2 - \alpha_{p-i}^2) \sum_{m=i}^{p-i-1} t^m.$$

Равенство (3.36) получается из этого выражения путем перегруппировки членов. \square

Замечание 30. Легко проверить, что остатки, которые были вычислены в ходе доказательств теорем 7 и 9, могут быть выведены из следствия 11. Для $AR(1)$ имеем $r_0(t) = 1 - \phi_1^2 = \alpha_0^2 - \alpha_1^2$. Для $AR(2)$ имеем $r_1(t) = -\phi_1(1 + \phi_2) = \alpha_0\alpha_1 - \alpha_1\alpha_2$ и $r_0(t) = (1 - \phi_2^2)(t + 1) = (\alpha_0^2 - \alpha_2^2)(t + 1)$.

Из следствия 11 следует, что одно из необходимых условий существования разложения $\mathbf{W} = \mathbf{A} * \mathbf{B}$ для случая диагональной матрицы \mathbf{B} с $B \geq p$ является $\phi_p = \pm 1$ (коэффициент перед t^{p-1} в (3.36) равен $\alpha_0^2 - \alpha_p^2 = 1 - \phi_p^2$ и должен быть равен 0). Это противоречит $|\phi_p| < 1$, необходимому условию стационарности $AR(p)$. Поэтому для матрицы \mathbf{W} , соответствующей стационарной модели $AR(p)$, деконволюция (3.10) не может быть выполнена.

3.3. Обнаружение сигналов с помощью сингулярного спектрального анализа Монте-Карло: Множественное тестирование

Вообще говоря, SSA — это метод, не требующий задания модели ряда. Тем не менее, значительная часть теории SSA посвящена извлечению компонентов временного ряда, которые управляются (возможно, приближенно) при помощи линейного рекуррентного соотношения; в частности, к этому классу сигналов относится сумма синусоид с медленно меняющимися амплитудами. SSA хорошо выделяет периодические компоненты, однако хорошо известно, что извлеченные компоненты могут быть ложными, поскольку они могут быть порождены шумом. В некотором смысле, это плата за непараметрическую природу метода. Если мы хотим применить статистический подход для принятия решения, необходимо предположить наличие модели. Обычно вопрос о существовании сигнала во временном ряде формулируется как проверка гипотезы о том, что ряд является стохастическим процессом. Критерий должен быть мощным против альтернативной гипотезы, которая утверждает существование неслучайного (например, периодического) компонента. Существует множество статистических критериев для проверки этих гипотез для различных классов стохастических процессов (см., например, [148] для красного шума или [149] для пространства состояний Орнштейна-Уленбека, которое является непрерывным аналогом авторегрессионных процессов первого порядка AR(1)). Мы рассматриваем построение такого критерия в рамках SSA, поскольку SSA способен восстанавливать обнаруженную компоненту временного ряда.

Из свойств SSA (см., например, описание связи между спектральной плотностью и собственными векторами в [1, Раздел 6.4.3], где обсуждаются результаты из [150] в терминах SSA) следует, что естественным предположением для модели шума является то, что шум красный (AR(1) с положительным коэффициентом). Это объясняется тем, что спектральная плотность красного шума монотонна; поэтому собственные векторы похожи на синусоиды и могут быть связаны с различными частотами. Напомним, что $\xi = (\xi_1, \dots, \xi_N)$ — красный шум с параметрами φ и δ , если $\xi_n = \varphi \xi_{n-1} + \delta \varepsilon_n$, где $0 < \varphi < 1$, ε_n — белый гауссовский шум со средним 0 и дисперсией 1, ξ_1 имеет нормальное распределение со нулевым средним и дисперсией $\delta^2/(1 - \varphi^2)$. Более того, в климатологии обычная модель представляет собой слабый сигнал (если он вообще есть) в красном шуме. Оба соображения вдохновили на создание метода Монте-Карло SSA (MC-SSA). MC-SSA был предложен в [151] и позже рассматривался во многих работах. ([152], [153], [154], [155], [156], [157], среди прочих).

Однако терминология в этих работах отличается от стандартной статистической терминологии и поэтому очень важно преодолеть разрыв между прикладным и статистическим подходами. Более того, это может помочь избежать неверных выводов в реальных приложениях. Изучение свойств критерия (и сравнение с другими тестами) будет проводиться путем оценки ошибок первого и второго рода.

Кратко опишем, как строится базовый тест MC-SSA. На шаге разложения SSA каждой компоненте разложения (связанной с собственными векторами автоковариационной матрицы

выборки, в базовом варианте) можно поставить в соответствие частоту, поскольку собственные векторы автоковариационной матрицы красного шума, которая является теплицевой матрицы Каца-Мердока-Сегё, являются синусоидами с почти равноудаленными частотами [158]. Собственные значения равны суммарным квадратам норм проекции столбцов траекторной матрицы на собственные векторы и поэтому отражают вклад компонентов разложения (и, следовательно, соответствующих частот) во временной ряд. Оригинальная идея MC-SSA заключается в оценке параметров красного шума и применении бутстреп-симуляции для построения интервалов предсказания для собственных значений в случае отсутствия сигнала. Если собственное значение временного ряда находится за пределами построенных интервалов предсказания, соответствующая частота собственного вектора считается значимой. Более того, можно восстановить обнаруженный сигнал. В модификациях MC-SSA выбор векторов для проектирования может варьироваться.

Из описания метода ясно, что существует проблема множественного тестирования, когда вероятность ложного обнаружения периодической компоненты для одной из рассматриваемых частот (групповая ошибка, family-wise error rate) неизвестна и много больше, чем заданный уровень значимости (single-test error rate). Эта проблема обсуждается в различных работах, посвященных MC-SSA. Теоретический подход к множественному тестированию, который мы предлагаем в этом разделе, позволяет построить многомерный критерий с заданной групповой ошибкой.

Новизна заключается в применении статистического подхода к проблеме обнаружения сигнала в рамках Монте-Карло SSA для контроля ошибки первого рода и оценки ошибки второго рода. Для одновременного тестирования наличия нескольких частот предложена множественная версия MC-SSA, позволяющая контролировать групповую ошибку. На основе численного исследования мы обсуждаем плюсы и минусы нескольких версий MC-SSA.

В разделе 3.3.2 представлены версии алгоритмов MC-SSA, включая предлагаемые алгоритмы множественного тестирования. Раздел 3.3.3 посвящен исследованию подхода MC-SSA с помощью статистического подхода.

3.3.1. Статистический подход к проверке гипотез

В работах, начиная с [151], метод MC-SSA описывается в основном как метод для решения прикладных задач, и поэтому способ описания не является обычным для статистики. Поэтому давайте начнем со статистического подхода к проблеме.

Пусть нулевой гипотезой является то, что временной ряд представляет собой чистый стационарный стохастический процесс. В рассматриваемом контексте это может быть белый или красный шум. Иногда говорят, что проверяется наличие сигнала в шуме, тогда как нулевая гипотеза формулируется как гипотеза об отсутствии сигнала в шуме. Рассмотрим критерий, который определяет, отвергается или не отвергается нулевая гипотеза. Если нулевая гипотеза отвергается при заданном уровне значимости α , то можно утверждать о наличии сигнала

(точнее, об отклонении от нулевой гипотезы). Вероятность отвергнуть нулевую гипотезу, если она верна, называется ошибкой первого рода (α_I). Если критерий правильный, то ошибка первого рода равна заданному уровню значимости (или, по крайней мере, не больше α). Различные критерии отличаются мощностью против альтернативной гипотезы. Мощность — это вероятность отклонить нулевую гипотезу, если альтернативная гипотеза верна. Альтернативная гипотеза о том, что временной ряд содержит периодическую составляющую, важна на практике; поэтому мы рассмотрим критерии, которые являются мощными против таких гипотез.

Если мы хотим, чтобы ложное открытие происходило редко, то выбираем небольшой уровень значимости α и это гарантирует, что мы отвергнем истинную нулевую гипотезу с вероятностью не больше α . Однако мы не можем выбрать очень маленький уровень значимости, так как мощность теста уменьшается с уменьшением α .

Заметим, что нельзя рассматривать критерий, если его ошибка первого рода превышает заданный уровень значимости (*радикальный* критерий). Поэтому, прежде чем сравнивать критерии по мощности, необходимо убедиться, что ошибка I рода лежит в заданном диапазоне. Если ошибка I рода меньше уровня значимости (*консервативный* критерий), то это допустимо, однако, это означает, что данный критерий может быть улучшен, то есть мощность может быть увеличена с помощью коррекции, чтобы ошибка I рода была ближе к уровню значимости.

Полезной характеристикой критерия является возможность интерпретации различия от нулевой гипотезы, если эта гипотеза отвергается.

Bootstrapping

Большинство критерии имеют следующий вид. Построенная тестовая статистика измеряет разницу между данными и нулевой гипотезой определенным образом. Существует порог такой, что если тестовая статистика больше порога, то нулевая гипотеза отвергается. Конечно, этот порог зависит от уровня значимости α . Нередко этот порог не может быть получен теоретически. Тогда используется имитационное моделирование. Суррогатные данные моделируются в соответствии с нулевой гипотезой, а тестовая статистика рассчитывается многократно (G) для определения порога; этот подход широко используется, см. например, [159]. Таким образом, находится порог, при котором нулевая гипотеза отвергается примерно αG раз. Оцененный порог используется для проверки гипотезы на реальных данных. Суррогатные данные должны быть получены точно так же, как тестовая статистика была получена для исходных данных. Описанный подход можно назвать подходом Монте-Карло. Этот подход позволяет построить критерий с ошибкой первого рода, стремящейся к α по мере того, как G стремится к бесконечности.

Однако подход Монте-Карло можно применять, если нулевая гипотеза полностью определяет данные. Например, нулевая гипотеза гласит, что временной ряд представляет собой

красный шум с дисперсией δ^2 и коэффициентом φ , где δ^2 и φ — известные числа.

К сожалению, на практике это не так. Поэтому используется так называемый бутстрэпинг (“подтяни себя за шнурки”). Если δ^2 и φ неизвестны, то они оцениваются с помощью реальных исследуемых данных, а затем суррогатные данные получаются путем имитации с оцененным параметром. Поскольку оцененные параметры отличаются от истинных (неизвестных) параметров, то ошибка первого рода может быть далека от α и тест может стать радикальным или консервативным. Таким образом, Монте-Карло SSA (так называется семейство конкретных алгоритмов) является разновидностью тестирования с бутстрэпингом.

Оценка ошибки первого рода и мощности

Приведенные выше рассуждения о связи между ошибкой первого рода, уровнем значимости α и мощностью не могут быть применены на практике, так как ошибка первого рода и мощность неизвестны. Если что-то неизвестно теоретически, то снова помогает моделирование.

Пусть построен критерий для принятия решения (отвергнуть или не отвергнуть) для заданного уровня значимости α . Он может быть построен теоретически или с помощью бутстрэпинга/симуляции внутри, это не имеет значения. Затем производится выборка данных с заданными параметрами в соответствии с нулевой гипотезой (это подход Монте-Карло) моделируются много раз (M). Затем определяется доля случаев с отклонением нулевой гипотезы, она является оценкой ошибки I рода.

Чтобы оценить мощность, мы должны установить альтернативную гипотезу. В выборе есть свобода. Общим правилом является включение в альтернативную гипотезу того случая, который является важным, т.е. случай, который следует отличать от нулевой гипотезы. Например, альтернатива может утверждать, что временной ряд представляет собой синусоиду с заданной частотой, амплитудой и фазой, искаженной шумом, с теми же параметрами, которые рассматривались для нулевой гипотезы.

Для оценки мощности выполняется процедура, аналогичная процедуре для ошибки I рода. Мы моделируем суррогатные данные с заданными параметрами в соответствии с альтернативной гипотезе много раз (M). Затем определяется доля случаев с отклонением нулевой гипотезы, являющаяся оценкой мощности.

Предсказательные интервалы для проверки гипотез

Пусть тестовая статистика t дает интерпретируемую характеристику данных (например, вклад частоты в наблюдаемый временной ряд). Таким образом, вопрос о наличии сигнала может быть переформулирован как “может ли данное значение t быть вызвано только шумовой составляющей?”.

Ответ на этот вопрос может быть получен стандартным способом. Поскольку вклад t является случайным, то для него существует предсказательный интервал. Предсказатель-

ный интервал может быть построен путем моделирования. Если мы сгенерировали выборку возможных вкладов, то интервал предсказания 95% — это интервал между 2,5% и 97,5% квантилями этой выборки. В статистической терминологии этот интервал не называется доверительным, поскольку доверительные интервалы строятся для неизвестных (неслучайных) параметров, и их длина будет стремиться к нулю, так как количество симуляций (размер выборки) стремится к бесконечности.

γ -Предсказательные интервалы служат для проверки гипотез с уровнем значимости α для $\alpha = 1 - \gamma$. Если наблюдаемое значение t не принадлежит предсказательному интервалу, то нулевая гипотеза отвергается. Удобно изобразить, скажем, 95%-ные предсказательные интервалы для t , чтобы визуализировать проверку гипотезы с уровнем значимости 5%.

Односторонние и двухсторонние критерии

Мы упоминали, что критерий состоит из тестовой статистики t и порога t_0 . Использование этого порога может быть различным. Более того, порог может состоять из двух чисел, t_1 и t_2 . Например, нулевая гипотеза может быть отвергнута, если тестовая статистика больше t_1 или меньше t_2 (двухсторонний тест) или если $t > t_0$ (односторонний тест). Выбор типа критерия зависит от альтернативной гипотезы, поскольку мы хотим увеличить мощность против выбранной альтернативы.

Если мы хотим обнаружить оба случая, когда вклад частоты либо больше, либо меньше, чем для чистого шума, то выбираем двухсторонний тест. Если мы хотим обнаружить только превышение вклада частоты, мы выбираем односторонний тест. Этот подход может быть выражен в терминах предсказательных интервалов. Если мы хотим найти частоту с вкладом, превышающим вклад шума, то односторонний предсказательный интервал имеет вид $[0, t_0]$ (в общем случае, $[-\infty, t_0]$; однако в нашем случае тестовая статистика неотрицательна). В двухстороннем случае двусторонний предсказательный интервал равен $[t_1, t_2]$.

Множественное тестирование

Проблема множественного тестирования хорошо известна. Описанный выше подход к статистическому тестированию применим только для одиночных тестов, поскольку вероятность ложного обнаружения контролируется только для каждого отдельного теста.

Если мы тестируем несколько тестов (m) одновременно, нас интересует так называемая групповая ошибка (FWER). FWER — это вероятность ложного обнаружения хотя бы в одном из m тестов. Эта вероятность может быть значительно больше, чем выбранное небольшое значение α . Таким образом, мы не должны использовать набор отдельных тестов с α , если хотим контролировать FWER. В идеале, мы должны построить один многомерный тест вместо нескольких одиночных тестов. Если это технически сложно, то используется поправка Бонферрони (проведение одиночных тестов с уровнем значимости α/m); этот трюк позволяет контролировать FWER не больше чем α (обычно при этом групповая ошибка первого

рода значительно меньше α ; то есть, множественное тестирование консервативно и, следовательно, снижает мощность теста). Если одиночные тесты независимы, то поправка Šidák (проведение одиночных тестов с уровнем значимости $1 - (1 - \alpha)^{1/m}$) дает точный тест; однако в общем случае тестирование с поправкой Šidák может быть радикальным, и поэтому поправка Šidák не применяется.

3.3.2. Монте-Карло SSA

Общие замечания

С одной стороны, Монте-Карло SSA является хорошо разработанным методом. В статье [151] основа метода подробно описана. С другой стороны, многие вопросы все еще находятся в стадии исследования; это, в частности, наилучший способ оценки параметров шума и учет наличия шумового сигнала. В этом разделе мы рассмотрим наиболее простую версию Монте-Карло SSA, чтобы сосредоточиться на проблеме множественного тестирования.

В некоторых случаях мы будем предполагать, что параметры шума известны. Также мы покажем, как могут изменяться ошибки критерия, если параметры оцениваются.

Напомним, что в SSA на первом этапе, в том числе, строится базис из левых сингулярных векторов $\{U_i\}$ траекторного пространства ряда. Векторы из этого базиса часто используются в Монте-Карло SSA для оценивания вклада частоты, соответствующей вектору. Иногда вместо Basic SSA рассматривается версия Toeplitz SSA, предназначенная для стационарных временных рядов. По свойствам SVD, равенство $\lambda_i = \|\Pi_{\{U_i\}} \mathbf{X}\|^2 = \|\mathbf{X}^T U_i\|^2 = \sum_{j=1}^K (X_j^T U_i)^2$ можно интерпретировать как суммарный квадрат нормы проекций запаздывающих векторов на $\text{span}(U_i)$.

Существует естественная модификация Монте-Карло SSA, которая не использует шаг разложения SSA, за исключением построения траекторной матрицы (часть исследования будет выполнена для такой модификации). Тем не менее, связь с SSA существенна, так как обнаруженный сигнал может быть восстановлен с помощью SSA, если мы сохраним связь с собственными векторами в SSA.

Одиночный тест

Пусть $\xi = (\xi_1, \dots, \xi_N)$ — красный шум с параметрами φ и δ . Обозначим через L длину окна, а через $\Xi = \mathcal{T}(\xi)$ траекторную матрицу ξ . Пусть задан вектор $W \in \mathbb{R}^L$, $\|W\| = 1$. Если нас интересует частотный вклад, то W может быть синусоидой с заданной частотой. Суммарный квадрат нормы проекции столбцов Ξ на вектор W вычисляется как

$$p = \|\Xi^T W\|^2 = W^T (\Xi \Xi^T) W.$$

Нулевая гипотеза утверждает, что наблюдаемый временной ряд X является реализацией ξ с некоторыми параметрами φ и δ . Обозначим $\hat{p} = \|\mathbf{X}^T W\|^2$. Если W — собственный

вектор $\mathbf{X}\mathbf{X}^T$, то \hat{p} — соответствующее собственное значение. Заметим, что для синусоиды W , \hat{p} практически не зависит от фазы этой синусоиды, так как при больших $K = N - L + 1$ запаздывающие векторы состоят из многих сдвигов.

Пусть известны ϕ и δ . При некоторых предположениях распределение p может быть вычислено теоретически. Тогда интервал прогноза с доверительной вероятностью γ рассчитывается как интервал между $(1 - \gamma)/2$ - и $(1 + \gamma)/2$ -квантилями для двухстороннего теста и между нулем и γ -квантилем для одностороннего теста (правостороннего). В обоих случаях \hat{p} принадлежит построенному прогнозируемому интервалу с вероятностью γ .

Если теоретическое распределение неизвестно, то эти квантили могут быть вычислены путем генерации G реализаций ξ^i случайного вектора ξ и использования эмпирических (выборочных) квантилей для полученной выборки $p_i = \|\mathbf{\Xi}_i^T W\|^2$, $i = 1, \dots, G$. Вероятность того, что \hat{p} принадлежит эмпирическому (Монте-Карло) интервалу предсказания, стремится к γ по мере того, как G стремится к бесконечности.

Напомним, что уровень значимости α равен $1 - \gamma$ и поэтому можно сказать, что вероятность ошибки первого рода α_I стремится к α .

Как для теоретических выкладок, так и при моделировании используются значения параметров ϕ и δ . Здесь мы не обсуждаем оценку параметров шума в присутствии сигнала. Отметим, что предложенный подход к проверке гипотез справедлив для различных модификаций метода MC-SSA.

Замечание 31. Важно отметить, что W может быть порожден самим временным рядом. Однако, как обсуждалось в [151], тогда суррогатные данные также должны быть спроектированы на произведенные ими векторы. Возникает соблазн рассмотреть вариант, когда вектор проекции создается наблюдаемым временным рядом, а тест строится путем проекции на этот вектор. Однако тогда ошибка первого рода не контролируется, а тест является радикальным.

Выбор векторов для проекции

На практике мы не знаем частоту возможной периодической составляющей сигнала. Поэтому подход заключается в том, чтобы рассмотреть множество векторов для проекции, которые соответствуют набору частот. Например, можно взять набор синусоид $W_1, \dots, W_H \in \mathbb{R}^L$ с равноудаленными частотами из некоторого интервала частот $[\omega_1, \omega_2] \subset (0, 0.5)$. Чтобы получить слабо зависимые тесты, число векторов не должно превышать их размерность L .

Другой вариант — взять набор собственных векторов, порожденных SSA (это обычный случай в MC-SSA) с учетом замечания 31.

Компромиссный вариант, предложенный в [151]: W_1, \dots, W_H — это собственные векторы ковариационной матрицы генеральной совокупности. То есть, матрица ковариации выборки $\mathbf{X}\mathbf{X}^T$, где \mathbf{X} — L -траекторная матрица, тогда как общепопуляционная корреляционная матрица красного шума имеет (i, j) -ый член $\phi^{|i-j|}$. Они близки, однако, с точки зрения ошибки

первого рода разница может быть кардинальной. Если в набор проекционных векторов необходимо включить вектор синусоиды с определенной частотой ω , то собственные векторы матрицы с (i, j) -ым членом $\phi^{|i-j|} + C \cos(2\pi\omega|i-j|)$ могут быть рассмотрены.

Таким образом, мы рассмотрим два варианта:

1. W_1, \dots, W_H — собственные векторы матрицы

$$\{\phi^{|i-j|} + C \cos(2\pi\omega|i-j|)\}_{i,j=1}^L. \quad (3.38)$$

2. W_1, \dots, W_H — собственные векторы матрицы $\mathbf{X}\mathbf{X}^T$.

Поскольку мы изучаем версию Монте-Карло SSA, где проекции суррогатных данных выполняются на фиксированные векторы, то вторая версия в общем случае неверна. Мы рассмотрим пример, чтобы показать это.

Множественное тестирование

В Монте-Карло SSA предсказательные интервалы строятся для вклада каждого вектора проекции независимо. Пусть W_1, \dots, W_H — набор проекционных векторов. Обозначим

$$\hat{p}_k = \|\mathbf{X}^T W_k\|^2, \quad k = 1, \dots, H.$$

Для каждого вектора W_k строится выборка квадратов норм проекций: $P_k = (p_{k1}, \dots, p_{kG})^T$, где p_{ki} вычисляется как

$$p_{ki} = \|\boldsymbol{\Xi}_i^T W_k\|^2, \quad i = 1, \dots, G; \quad (3.39)$$

здесь $\boldsymbol{\Xi}_i = \mathcal{T}(\boldsymbol{\xi}^i)$ — траекторная матрица i -ой выборки красного шума $\boldsymbol{\xi}^i = (\xi_1^i, \dots, \xi_N^i)$ с заданными параметрами.

Мы можем построить одиночные предсказательные интервалы для вклада каждого вектора W_k , как это описано в разделе 3.3.2. Проблема множественного тестирования (проблема FWER, который может быть намного больше заданного уровня значимости α) может быть решена с помощью поправки Бонферрони. Однако при использовании поправки Бонферрони тест становится консервативным (FWER меньше α). Для получения точного теста используется подход, аналогичный HSD Тьюки, применяемый к post hoc сравнениям в ANOVA. То есть, мы можем построить тест, который основан на распределении максимума стандартизированного вклада p_k . Если этот тест отвергает нулевую гипотезу, то все частоты, соответствующие проекционным векторам с вкладом, лежащим вне скорректированных интервалов предсказания, считаются значимыми. Таким образом, о значимости частоты можно говорить, если поставить в соответствие проекционные векторы и частоты. Заметим, что для данного подхода FWER равен α .

Опишем алгоритм построения предсказательных интервалов с поправкой на множественное тестирование.

Первый вариант прост, это одиночный тест для каждого вектора W_k с поправкой Бонферрони, то есть вместо α берется уровень значимости α/H .

АЛГОРИТМ 3.4: Одиночный односторонний тест с поправкой Бонферрони

- 1: Для каждого k , $k = 1, \dots, H$, вычислить тестовую статистику \hat{p}_k и выборку $P_k = \{p_{ki}\}_{i=1}^G$, см. (3.39).
- 2: Найти q_k как выборочный $(1 - \alpha/H)$ -квантиль для P_k .
- 3: Нулевая гипотеза, утверждающая, что временной ряд является чистым красным шумом, не отвергается, если для каждого k справедливо неравенство $\hat{p}_k < q_k$; в противном случае нулевая гипотеза отвергается и обнаруживается сигнал.
- 4: Если H_0 отвергается, то можно провести post-hoc тестирование: вклад W_k (и соответствующей частоты) является значимым, если \hat{p}_k превышает q_k . Таким образом, $[0, q_k]$ рассматриваются как скорректированные предсказательные интервалы, $k = 1, \dots, H$.

АЛГОРИТМ 3.5: Одиночный двухсторонний тест с поправкой Бонферрони

- 1: Для каждого k , $k = 1, \dots, H$, вычислить тестовую статистику \hat{p}_k и выборку $P_k = \{p_{ki}\}_{i=1}^G$, см. (3.39).
- 2: Найти q_k^{low} и q_k^{up} как выборочные $(0.5\alpha/H)$ - и $(1 - 0.5\alpha/H)$ -квантили соответственно.
- 3: Нулевая гипотеза, утверждающая, что временной ряд является чистым красным шумом, не отвергается, если для каждого k $q_k^{\text{low}} < \hat{p}_k < q_k^{\text{up}}$; в противном случае нулевая гипотеза отвергается и обнаруживается сигнал.
- 4: Если H_0 отвергнута, то можно провести post-hoc тестирование: вклад W_k (и соответствующей частоты) значителен, если \hat{p}_k не принадлежит $[q_k^{\text{low}}, q_k^{\text{up}}]$. Таким образом, $[q_k^{\text{low}}, q_k^{\text{up}}]$ рассматриваются как скорректированные предсказательные интервалы, $k = 1, \dots, H$.

Расширим подход множественного тестирования, который был рассмотрен в [160] в рамках Монте-Карло SSA. Подход основан на распределении $\max_{1 \leq k \leq H} (p_{ki} - \mu_k)/\sigma_k$, где μ_k и σ_k — среднее и стандартное отклонение P_k , $k = 1, \dots, H$. Здесь σ_k отражает размер k -го предсказательного интервала. Нормализация по σ_k сохраняет ту же разницу между размерами интервалов, что и в одиночных тестах. По построению, размер предсказательного интервала связан с мощностью теста против существования сигнала на соответствующей частоте. Поэтому можно положить $\tilde{\sigma}_k = w_k \sigma_k$, где w_k — вес; увеличение веса соответствует большему ожиданию сигнала на частоте, соответствующей W_k . Некоторые веса w_k могут быть нулевыми; это означает, что соответствующие проекционные векторы не участвуют в тестировании и тест не способен обнаружить сигнал на соответствующих частотах. Таким образом, все

равно, задавать нулевые веса вне частотного диапазона (и единичные веса в нем) или просто рассматривать подмножество проекционных векторов, соответствующих частотному интервалу. Однако, подход с весами гораздо более гибкий и позволяет брать произвольные веса между 0 и 1.

АЛГОРИТМ 3.6: Взвешенный множественный односторонний тест

- 1: Для каждого k , $k = 1, \dots, H$, вычислить тестовую статистику \hat{p}_k , выборку $P_k = \{p_{ki}\}_{i=1}^G$, см. (3.39), и вычислить ее среднее μ_k и стандартное отклонение σ_k .
- 2: Вычислить $\boldsymbol{\eta} = (\eta_1, \dots, \eta_G)$, где

$$\eta_i = \max_{1 \leq k \leq H} (p_{ki} - \mu_k) / \tilde{\sigma}_k, \quad \tilde{\sigma}_k = w_k \sigma_k, \quad i = 1, \dots, G.$$

- 3: Найти q как выборочный $(1 - \alpha)$ -квантиль от $\boldsymbol{\eta}$.
- 4: Нулевая гипотеза, утверждающая, что временной ряд является чистым красным шумом, не отвергается, если

$$\max_{1 \leq k \leq H} (\hat{p}_k - \mu_k) / \tilde{\sigma}_k < q;$$

в противном случае, нулевая гипотеза отвергается и обнаруживается сигнал.

- 5: Если H_0 отвергается, то можно провести post-hoc тестирование: вклад W_k (и соответствующей частоты) является значимым, если \hat{p}_k превышает $\mu_k + qw_k\sigma_k$. Таким образом, $[0, \mu_k + qw_k\sigma_k]$ рассматриваются как скорректированные интервалы предсказания, $k = 1, \dots, H$.
-

АЛГОРИТМ 3.7: Взвешенный множественный двухсторонний тест

- 1: Для каждого k , $k = 1, \dots, H$, вычислить тестовую статистику \hat{p}_k , выборка $P_k = \{p_{ki}\}_{i=1}^G$, см. (3.39), и вычислить ее среднее μ_k и стандартное отклонение σ_k .
- 2: Вычислить $\boldsymbol{\eta} = (\eta_1, \dots, \eta_G)$, где

$$\eta_i = \max_{1 \leq k \leq H} |p_{ki} - \mu_k| / \tilde{\sigma}_k, \quad \tilde{\sigma}_k = w_k \sigma_k, \quad i = 1, \dots, G.$$

- 3: Найти q как выборочный $(1 - \alpha)$ -квантиль от $\boldsymbol{\eta}$.
- 4: Нулевая гипотеза, утверждающая, что временной ряд является чистым красным шумом, не отвергается, если

$$\max_{1 \leq k \leq H} |\hat{p}_k - \mu_k| / \tilde{\sigma}_k < q;$$

в противном случае, нулевая гипотеза отвергается и обнаруживается сигнал.

- 5: Если H_0 отвергается, то можно провести post-hoc тестирование для фиксированного k : вклад W_k (и соответствующей частоты) является значимым, если $|\hat{p}_k - \mu_k|/\tilde{\sigma}_k$ превышает q . Таким образом, $[\mu_k - qw_k\sigma_k, \mu_k + qw_k\sigma_k]$ рассматриваются как скорректированные предсказательные интервалы, $k = 1, \dots, H$.
-

Алгоритмы 3.5 и 3.7 соответствуют двусторонним предсказательным интервалам (это обычная версия Монте-Карло SSA), в то время как алгоритмы 3.4 и 3.6 соответствуют одностороннему случаю. Это соответствует гиперпрямоугольным и полугиперпрямоугольной областям предсказания соответственно.

Замечание 32. Выбор векторов W_k , $k = 1, \dots, H$, обсуждался в разделе 3.3.2. Для увеличения мощности критерия число H векторов должно быть как можно меньше, например, можно взять только векторы с доминирующими частотами из заданного диапазона частот. Если векторы W_k — синусоиды, то выбор векторов с частотами из заданного диапазона тривиален. Если W_k — собственные векторы, то их доминирующие частоты можно вычислить, например, методом ESPRIT [30, Раздел 3.1]. Это не имеет смысла для белого шума, для которого каждый собственный вектор представляет собой смесь множества частот. Однако для красного шума это имеет смысл, поскольку собственные векторы соответствуют узким диапазонам частот. Как мы уже упоминали выше, увеличение мощности может быть выполнено путем установки весов.

3.3.3. Численное исследование

Введем аббревиатуры для версий теста. Мы рассматриваем 3-символьные аббревиатуры с необязательной информацией

$$\{M,S,B\}\{T,E\}\{1,2\}[Est]. \quad (3.40)$$

Первая буква M (Multiple) соответствует алгоритмам 3.6 и 3.7, S (Single) — алгоритмам 3.4 и 3.5 без поправки Бонферрони, а B означает использование этой поправки. Вторая буква связана со способом генерации проекционных векторов: T означает, что рассматриваются собственные векторы теоретической ковариационной матрицы (3.38) красного шума ($C = 0$); E означает, что используются собственные векторы эмпирической ковариационной матрицы $\mathbf{X}\mathbf{X}^T$. Последняя цифра означает, какой тест (односторонний или двухсторонний) рассматривается. В качестве основного мы рассматриваем метод MT1, который является точным и более мощным.

Введенные аббревиатуры соответствуют использованию истинных параметров красного шума и единичных весов w_k . Если параметры шума оцениваются, то после цифры добавляем ‘Est’. Если используется другая модификация, то это указывается отдельно.

R-скрипты в [161] содержат реализацию приведенных выше алгоритмов.

Продемонстрируем результаты различных версий Монте-Карло SSA.

Пример

Модель искусственного временного ряда имеет вид

$$x_n = A \sin(2\pi\omega n) + \xi_n, \quad (3.41)$$

где ξ_n — красный шум с параметрами φ и δ , $n = 1, \dots, N$. Случай $A = 0$ соответствует нулевой гипотезе, а случай $A > 0$ говорит о наличии сигнала, то есть соответствует альтернативе. Здесь и далее мы рассматриваем параметры AR(1) $\varphi = 0.7$ и $\delta = 1$.

Для иллюстрации возьмем $N = 1000$ и параметры сигнала $A = 0.4$ и $\omega = 0.2$ в H_1 . Параметры MC-SSA — $L = 40$ и $G = 1000$.

Рассмотрим фиксированный уровень значимости 0.2 (то есть доверительная вероятность равна 0.8). Чтобы ослабить зависимость от длины временного ряда N , считаем $\|\mathbf{X}\|^2 = \sum_{i=1}^N x_i^2 / N$. Для создания суррогатных данных мы использовали истинные параметры AR(1). Непрерывная кривая это спектральная плотность AR(1) с параметрами, которые использовались в моделировании. Доминирующие частоты W_k мы вычислили методом ESPRIT с рангом $r = 2$.

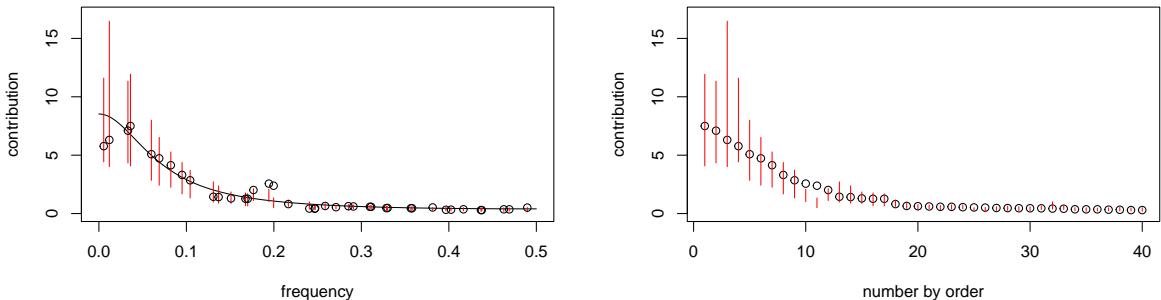


Рис. 3.4: МЕ2. Слева — упорядочены по частоте, справа — упорядочены по значению проекции (\hat{p}_k).

Хотя односторонние версии тестов являются более мощными, на большинстве рисунков мы показываем двухсторонние интервалы предсказания для более наглядного представления.

Наличие синусоидального сигнала в основном соответствует превышению верхней границы. Поэтому, как правило, для точных тестов, если верхняя граница меньше, то тест более мощный.

Рис. 3.4 демонстрирует, вообще говоря, радикальный тест, который поэтому не может быть использован на практике как есть. В этой версии проекционные векторы генерируются

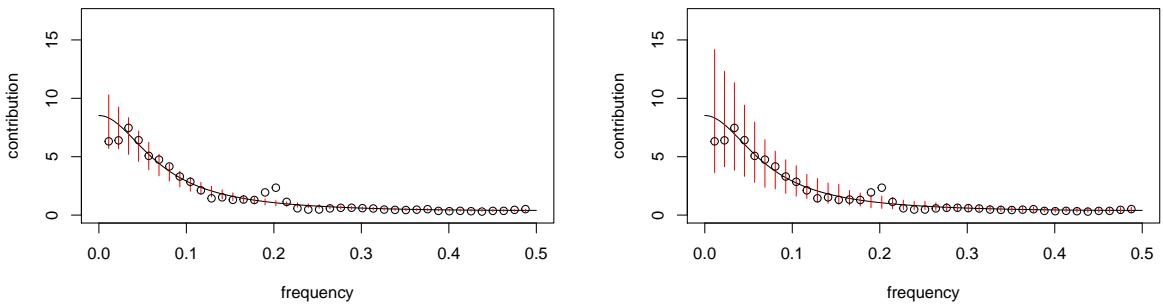


Рис. 3.5: $\{S,B\}T2$. Слева — без коррекции, справа — с поправкой Бонферрони.

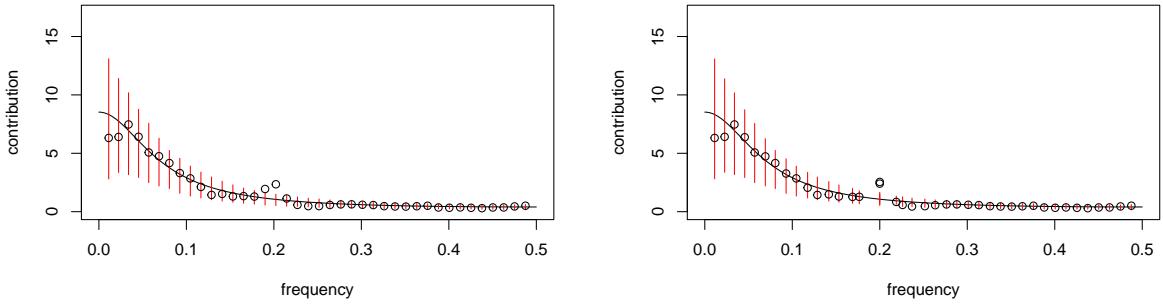


Рис. 3.6: $MT2$. Слева — собственные векторы ковариационной матрицы чистого красного шума, справа — собственные векторы ковариационной матрицы красного шума с добавленной синусоидой на частоте 0.2.

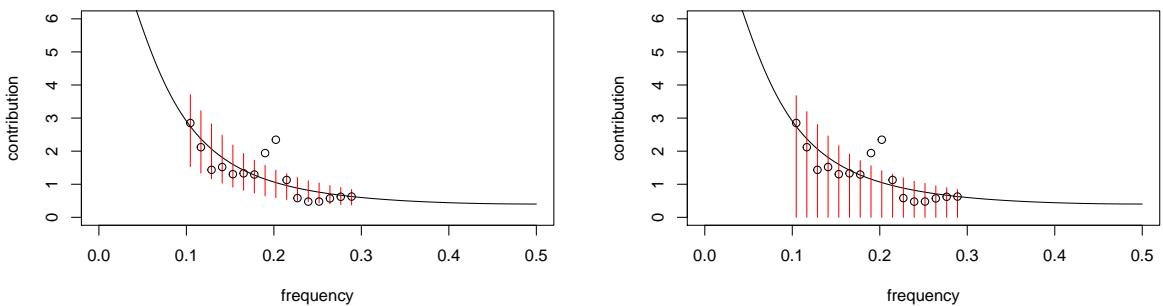


Рис. 3.7: $MT\{1,2\}$, диапазон частот $[0, 1, 0, 3]$. Слева — двухсторонний тест, справа — односторонний тест.

наблюдаемым временным рядом; этот выбор соответствует популярной версии Монте-Карло SSA.

На рис. 3.5 показаны предсказательные интервалы для одиночного теста с проекцион-

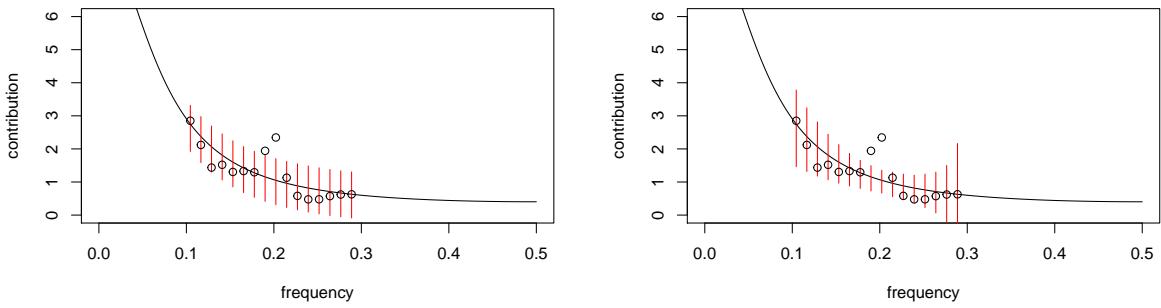


Рис. 3.8: МТ2, взвешенный, диапазон частот $[0.1, 0.3]$, двухсторонний тест. Слева — веса для получения равных интервалов предсказания ($w_k = 1/\sigma_k$), справа — веса для увеличения мощности против частот около 0.2.

ными векторами, выбранными как теоретические собственные векторы; на рис. 3.5 (справа) используется поправка Бонферрони. Тест на рис. 3.5 (слева) является сильно радикальным, а на рис. 3.5 (справа) слегка консервативным.

Рис. 3.6 содержит несколько предсказательных интервалов, которые обеспечивают точный тест с теоретическими собственными векторами в качестве проекционных векторов. На рис. 3.6 (слева) в качестве W_k взяты собственные векторы матрицы (3.38) с $C = 0$. Верхние границы немного ниже, чем на рис. 3.5 (слева). На рис. 3.6 (справа) используется информация о наличии сигнала с частотой $\omega = 0.2$, а в качестве W_k взяты собственные векторы матрицы (3.38) с $C = 1$.

Рис. 3.7 и 3.8 относятся к случаю нулевых весов для частот проекционных векторов вне частотного интервала $[0.1, 0.3]$. Рис. 3.7 демонстрирует разницу между двухсторонними и односторонними интервалами предсказания. В одностороннем случае верхняя граница немногого ниже. На рис. 3.8 показано, как веса влияют на размеры предсказательных интервалов. Мы установили `weights = c(seq(6.25, 8, 0.25), 8:1)`, что означает, что веса увеличиваются от 6.25 до 8 с шагом 0.25, а затем уменьшаются от 8 до 1. Влияние весов на мощность теста будет изучено позже.

Исследование статистических свойств MC-SSA

Опишем методику исследования статистических свойств построенных критериев с помощью бутстреппинга. Ключевыми пунктами являются:

1. Первый шаг — смоделировать синтетические данные в соответствии с нулевой гипотезой M раз и оценить $\alpha_l = \alpha_l(\alpha)$ как долю отвергнутых нулевых гипотез для заданного уровня значимости α . Проверить, что выполняется необходимое условие $\alpha_l \leq \alpha$. Необходимо найти достаточный размер G суррогатных данных.

2. Если необходимое условие выполнено, то вторым шагом будет моделирование синтетических данных в соответствии с альтернативной гипотезой M раз и оценка мощности $1 - \alpha_{II}$ против этой гипотезы как доли отвергнутых нулевых гипотез для данного уровня значимости α .
3. Сравнить различные критерии по мощности против интересующей альтернативной гипотезы и использовать тот, который имеет большую мощность.
4. Если не существует теста с достаточной мощностью, то можно улучшить тест, где $\alpha_I < \alpha$ или $\alpha_I > \alpha$. Тогда найти $\tilde{\alpha}$ такой, что $\alpha_I(\tilde{\alpha}) \approx \alpha$, и использовать уровень значимости $\tilde{\alpha}$ вместо α .

Ниже мы рассмотрим эти пункты. Как и раньше, мы рассматриваем параметры $AR(1)$ $\varphi = 0.7$ и $\delta = 1$. В случае, когда оцениваются параметры $AR(1)$, применялся метод максимального правдоподобия, где для нахождения начальных значений использовался метод условной суммы квадратов.

Выбор G . Достаточное значение G в пункте 1 может быть определено теоретически. Значение G должно быть значительно большим, чтобы оценить квантили для суррогатных данных. В случае одиночных предсказательных интервалов с поправкой Бонферрони необходимо оценить $(1 - \alpha/H)$ -квантили (будем рассматривать односторонние тесты), в то время как в случае множественного тестирования нам нужны только $(1 - \alpha)$ -квантили. Например, если $\alpha = 0.05$, то для множественного тестирования достаточно $G = 1000$, так как оценка является 50-м значением от максимума в упорядоченной выборке. Однако, для одиночного тестирования и $H = 100$ проекционных векторов, $\alpha/100 = 0.005$ и поэтому 0.995-квантиль будет недооценен (и, следовательно, тест будет радикальным). Таблица 3.1 содержит оценки ошибок первого рода для $N = L = 200$ и $\alpha = 0.2$ (тогда поправка Бонферрони дает уровень значимости 0.001); использовалось $M = 10000$ симуляций. Таблица 3.1 подтверждает, что множественные методы требуют меньшего G . Здесь и далее столбцы ‘2.5%’ и ‘97.5%’ показывают границы 95% доверительных интервалов для оцененных вероятностей (ошибки I рода или мощности).

Таблица 3.1: Ошибка I рода для разных G .

Метод	G	расчетная ошибка первого рода	2,5%	97,5%
MT1	100	0.466	0.456	0.476
BT1	100	0.864	0.857	0.870
MT1	500	0.250	0.241	0.258
BT1	500	0.326	0.316	0.335
MT1	1000	0.219	0.211	0.227
BT1	1000	0.212	0.204	0.221

Ошибка первого рода. Оценки групповых ошибок I рода приведены на рис. 3.9; мы используем $N = 100$, $L = 10$, $M = 1000$ и $G = 1000$.

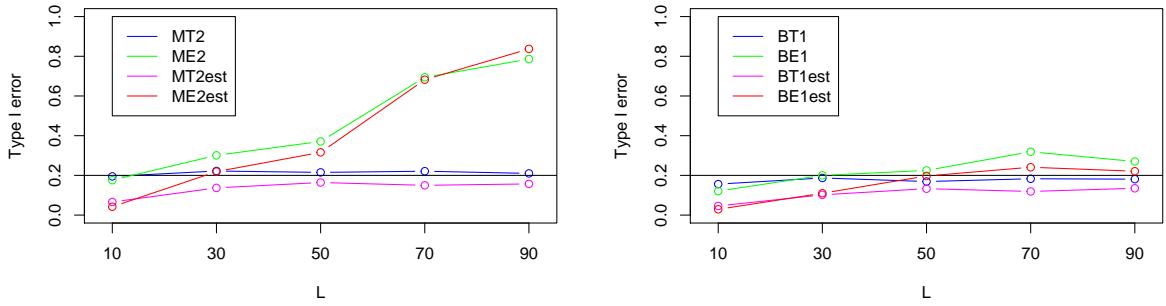


Рис. 3.9: Ошибка I рода для различных L , эмпирических и теоретических собственных векторов. Слева — множественный тест, справа — одиночный тест с поправкой Бонферрони

Рис. 3.9 (слева) демонстрирует разницу между использованием теоретических и эмпирических собственных векторов. Плюсы и минусы этих версий следующие.

1. Множественный тест, эмпирические собственные векторы.

Плюс: Можно восстановить сигнал на основе значимых проекционных векторов (эмпирических собственных векторов).
Минус: Тест является радикальным.

2. Множественный тест, теоретические собственные векторы.

Плюс: тест точный.

Минус: Если необходимо выполнить SSA восстановление обнаруженнего сигнала, необходимо провести дополнительный анализ, чтобы найти соответствие между теоретическими и эмпирическими собственными векторами.

На рис. 3.9 (справа) показана разница между использованием нескольких и одного теста.

Плюсы и минусы этих версий следующие.

1. Множественный тест

Плюс: тест является точным.

Минус: Пересчет необходим для рассмотрения подмножества проекционных векторов.

2. Одиночный тест

Плюс: Легко рассмотреть подмножества проекций векторов без пересчета.

Минус: Для версии без коррекции это очень радикальный тест. Для версии с поправкой Бонферрони, это немного консервативный тест (это заметно для малых L) и требуется большое G (заметно для больших L).

Кроме того, рис. 3.9 показывает, что если параметры шума оценены, то тесты МТ и ВТ становятся очень консервативными (ошибка первого рода становится значительно меньше заданного уровня $\alpha = 0.2$). Кроме того, консервативность поправки Бонферрони может компенсировать радикальность теста $\{\text{M,B}\}\text{E}$; это может быть причиной того, что тест ВЕ менее радикален, чем МЕ на рис. 3.9.

Мощность. Давайте оценим мощность различных версий SSA-тестов Монте-Карло. Рассмотрим наличие сигнала в H_1 с $A = 1$ и изучим зависимость мощности теста от частоты сигнала. Мы использовали стандартный метод оценки параметров AR(1), который игнорирует возможное присутствие сигнала.

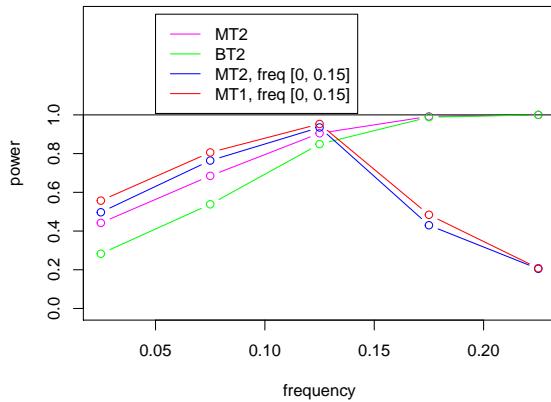


Рис. 3.10: Мощность для альтернатив с синусоидальными сигналами различных частот.

Оценки мощности изображены на рис. 3.10. Видно, что и односторонняя версия, и узкий частотный интервал могут увеличить мощность теста.

Коррекция ошибки первого рода. Проиллюстрируем пункт 4 описанной схемы на примере (3.41), где H_0 соответствует $A = 0$, а H_1 соответствует $A = 1$, $\omega = 0.1$. Как и ранее, мы рассматриваем фиксированный уровень значимости 0.2.

Если параметры шума оцениваются, то критерий становится очень консервативным (ошибка первого рода в два раза меньше, чем при заданном $\alpha = 0.2$). Изменяя α , можно обнаружить, что для $\tilde{\alpha} = 0.35$ мы получим $\alpha_l \approx 0.2 = \alpha$ (таблица 3.2, первые три строки).

Критерий со скорректированным формальным уровнем значимости является более мощным, чем первоначальный консервативный критерий. Последние три строки таблицы 3.2 показывают, что использование консервативного теста MT1est снижает мощность по сравнению с MT1, а после коррекции уровня значимости до $\tilde{\alpha} = 0.35$ мощность MT1est значительно возрастает.

Обратим внимание, что зависимость мощности теста от уровня значимости показывает мощность теста против выбранной альтернативной гипотезы. Чтобы выяснить, как будет работать тест, если формальный уровень значимости скорректировать для получения заданной вероятности ошибки первого рода, следует изучить зависимость мощности теста от вероятности ошибки первого рода; эта зависимость является своего рода ROC-кривой.

Таблица 3.2: Увеличение мощности консервативного теста.

	расчетная ошибка первого рода	2,5%	97,5%
MT1	0.200	0.176	0.226
MT1est($\tilde{\alpha} = 0.2$)	0.069	0.054	0.086
MT1est($\tilde{\alpha} = 0.35$)	0.208	0.183	0.234
	расчетная мощность	2.5%	97.5%
MT1	0.800	0.774	0.824
MT1est($\tilde{\alpha} = 0.2$)	0.575	0.544	0.606
MT1est($\tilde{\alpha} = 0.35$)	0.731	0.702	0.758

3.4. Выбор параметров

3.4.1. Введение

Постановки задач в рамках SSA могут различаться по следующим аспектам.

- A1 Предмет интереса: нас может интересовать либо сигнал S_N в целом, либо некоторые его характеристики. В частности, если S_N имеет конечный ранг, то он имеет известную параметрическую форму и мы можем быть заинтересованы в оценке параметров. Наиболее разработанной задачей является оценка коэффициентов затухания и частот экспоненциально затухающих синусоид в шуме. Для решения этой задачи оценки достаточно знать только r ведущих собственных векторов в (1.12), а точнее, подпространство, охватываемое этими собственными векторами (т.е. оцениваемое подпространство сигнала S_N); см. например, ESPRIT-подобные методы.
- A2 Тип остатков: остаток R_N является либо детерминированным, либо стохастическим (или он имеет как случайные, так и детерминированные компоненты). Эти случаи соответствуют различным свойствам разложения SSA и вызывают различные характеристики оценок. Например, ряд S_N конечного ранга будет ведущей компонентой, если детерминированный остаток ограничен некоторой константой, в то время как белый шум может иметь любую дисперсию при большой длине временного ряда N и длине окна L . Кроме того, структура стохастического шума (например, белый или красный) может влиять на поведение ошибок оценки.
- A3 Выбор длины окна: либо мы можем варьировать длину окна L , либо L фиксирована. В первом случае возникает проблема оптимального выбора L . Тогда асимптотическое поведение зависит от того, стремится ли L к бесконечности при $N \rightarrow \infty$ или нет. Если мы рассматриваем матрицу $L \times K$ с фиксированным числом строк в качестве входных данных, то единственный способ — это зафиксировать L и рассматривать K , стремящуюся к бесконечности. Другая возможная причина выбора не очень большого L — это вычислительные затраты. Однако в последнее время появились новые вычислительные приёмы, которые делают вычисления очень быстрыми, см. [113].

Следуя работе [5], мы рассмотрим различные постановки задачи в рамках аспекта A1, а затем проанализируем ошибки и правила выбора параметров в рамках аспектов A2 и A3.

Основная информация о структуре временного ряда, которую мы получаем в рамках SSA, содержится в наборе собственных троек $(\sqrt{\lambda_i}, U_i, V_i)$. Следовательно, мы получаем не только восстановленный сигнал, но и много дополнительной информации о S_N . Помимо задачи восстановления S_N , мы рассматриваем задачи прогнозирования сигнала и оценки его параметров.

Заметим, что SVD определяется только набором собственных векторов U_i , поскольку $\lambda_i = \|\mathbf{X}^T U_i\|^2$ и $V_i = \mathbf{X}^T U_i / \sqrt{\lambda_i}$, где \mathbf{X} — траекторная матрица наблюдаемого временного ряда.

Следовательно, $\tilde{\mathbf{S}} = \sum_{i=1}^r U_i U_i^T \mathbf{X} = \mathbf{P}_r \mathbf{X}$, где \mathbf{P}_r — ортогональная проекция на \mathcal{L}_r . Поэтому можно сказать, что набор собственных векторов (вместе с временным рядом X_N) полностью определяет все SVD-разложение и, следовательно, результаты прогнозирования и оценки параметров. Поэтому естественно начать исследование с оценки собственных векторов или, эквивалентно, с оценки собственного пространства. Отметим, что задача оценки факторных векторов V_i превращается в задачу оценки собственных векторов U_i при изменении длины окна с L на $N - L + 1$.

Отметим, что преобразования, применяемые к собственным векторам для получения, например, оценки частоты могут сильно изменить структуру ошибок оценки.

Тем не менее, в разделе 3.4.2 мы рассматриваем ошибки оценки подпространства сигнала как отправную точку исследования. Разделы 3.4.3 и 3.4.4 содержат результаты по восстановлению и прогнозированию на основе выбранного подпространства. Раздел 3.4.5 посвящен оценке параметров в рамках методов обработки сигналов на основе подпространств, включая ESPRIT. В разделах 3.4.2–3.4.5 мы даем общие рекомендации по выбору длины окна L , которые основаны на моделировании и известных теоретических результатах. В разделе 3.4.6 мы рассматриваем скорость сходимости в различных условиях (фиксированная длина окна L или длина окна пропорциональна длине временного ряда) для задач, исследованных в предыдущих разделах.

Отметим, что результаты разделов 3.4.2–3.4.6 справедливы при условии сильной разделимости сигнала и остатка. В разделе 3.4.7 рассматривается несколько примеров, в которых сильная разделимость отсутствует. В разделе 3.4.8 мы рассматриваем несколько версий Basic SSA и демонстрируем некоторые примеры применения версий, разработанных для стационарных временных рядов, к нестационарным.

В разделе 3.4.9 мы кратко описываем некоторые истоки SVD, обеспечивающего ключевой шаг алгоритма SSA. Нас интересуют эти истоки, поскольку они подразумевают различные взгляды на постановку задачи и на выбор параметров.

3.4.2. Сигнальное подпространство

В целом мы будем опираться на результаты работы [162], которая посвящена обсуждению сходимости, а также содержит основные члены ошибки и их верхние границы.

В качестве меры ошибки для аппроксимации подпространства мы рассматриваем спектральную норму разности проекторов на истинное подпространство и оцениваемое подпространство. Заметим, что эта норма равна синусу наибольшего главного угла между этими подпространствами. Цель данного раздела — исследовать зависимость ошибки аппроксимации от длины окна.

Пусть S_N — сигнал ранга r . За $P_r^{(s)}$ обозначим ортогональный проектор на подпространство сигнала $\mathcal{L}_r^{(s)}$, которое порождается левыми сингулярными векторами $U_1^{(s)}, \dots, U_r^{(s)}$ траекторной матрицы сигнала \mathbf{S} , а через \mathbf{P}_r мы обозначаем ортогональный проектор на под-

пространство оцененного сигнала $\mathcal{L}_r = \text{span}(U_1, \dots, U_r)$, где U_1, \dots, U_r — это r ведущих левых сингулярных векторов траекторной матрицы наблюдаемого временного ряда X_N . Заметим, что мы можем легко вычислить ошибку оценки $\|\mathbf{P}_r^{(s)} - \mathbf{P}_r\|$, поскольку косинус наибольшего главного угла между $\mathcal{L}_r^{(s)}$ и \mathcal{L}_r равен r -му собственному значению матрицы $\mathbf{U}_r^{(s)} \mathbf{U}_r^T$, где $\mathbf{U}_r^{(s)} = [U_1^{(s)} : \dots : U_r^{(s)}]$ и $\mathbf{U}_r = [U_1 : \dots : U_r]$ (см., например, [163, стр. 18]).

Рассмотрим пять примеров временных рядов $x_n = s_n + r_n$, $n = 1, \dots, N$,

$$s_n = 1, \quad r_n = -c(-1)^n, \quad (3.42)$$

$$s_n = b^n \cos(2\pi n/10), \quad r_n = c, \quad (3.43)$$

$$s_n = b^n \cos(2\pi n/10), \quad r_n = \sigma \varepsilon_n, \quad (3.44)$$

$$s_n = b^n \cos(2\pi n/10), \quad r_n = (\sigma \varepsilon_n + c)/\sqrt{2}, \quad (3.45)$$

$$s_n = b^n \cos(2\pi n/10), \quad r_n = \sigma \eta_n. \quad (3.46)$$

Здесь ε_n — белый гауссовский шум с дисперсией 1, а η_n — авторегрессионный процесс порядка 1 (красный шум) с параметром α и дисперсией $\mathbf{D}\eta_n = 1$, то есть $\eta_n = \alpha \eta_{n-1} + \varepsilon_n$, где ε_n имеет дисперсию $1 - \alpha^2$. В этом разделе мы установили $c = \sigma = 0.1$, $\alpha = 0.5$, $b = 1$, и $N = 100$.

Мы выбираем уровень шума во временных рядах (3.43)–(3.46), чтобы иметь одинаковое отношение сигнал/шум (SNR), которое обычно определяется как отношение среднего квадрата значений сигнала к среднему квадрату остаточных значений (или к дисперсии остатков, если они случайны).

Как правило, SNR не определяет величину ошибок оценок, полученных при обработке SSA, поскольку SNR не учитывает длину временного ряда. Фактически, SNR может быть использован для сравнения качества обработки временных рядов одинаковой длины. Однако мы не можем сказать, что SSA разделяет сигнал и шум только если SNR больше определенного значения; например, при любом малом SNR синусоидальный сигнал асимптотически отделяется от белого шума при $N \rightarrow \infty$ и $L \sim \beta N$, $0 < \beta < 1$.

Временной ряд (3.42) включен в рассмотрение, чтобы сравнить результаты с результатами в [162, Раздел 4.2.1]. Оказывается, что главный член возмущения, найденный в [162], почти равен всей погрешности. Более того, поведение ошибок зависит от того, являются ли длины окна и временного ряда четными или нечетными. Кроме того, это пример временного ряда, который дает ошибку проектора с членом первого порядка относительно уровня возмущения, который не является главным членом ошибки при $N \rightarrow \infty$.

Временные ряды (3.43)–(3.45) отличаются по структуре остатков: детерминированные, случайные или комбинированные. Временной ряд (3.46) используется для рассмотрения шума, отличающегося от белого шума.

В случае случайных остатков мы вычисляем либо MSD (среднее квадратическое отклонение), либо RMSE (квадратный корень из средней квадратической ошибки) в качестве меры точности. Как правило, эти критерии дают очень похожие результаты. Разница между ними

заключается в том, что мы можем вычислить квадратный корень до усреднения (MSD) или после усреднения (RMSE) для результатов моделирования. В примерах этого раздела мы оцениваем MSD, используя 100 моделей рядов.

На рис. 3.11–3.14 представим результаты моделирования для временных рядов, заданных формулами (3.42)–(3.46). На рис. 3.11 показаны ошибки в оценке проектора $P_r^{(s)}$ на сигнальное подпространство для примеров с детерминированными остатками. Можно заметить тенденцию к увеличению ошибок при увеличении длины окна. Однако при длине окна L , кратной периодам компонентов временного ряда, ошибки как правило, уменьшаются. Это отражает влияние кратности L и/или K периодам компонентов временного ряда. Заметим, что если и L , и K кратны периодам (на 2 в первом примере и на 10 во втором), то возмущение проектора равно 0. Это соответствует случаю биортогональности траекторных матриц рядов S_N и R_N и, следовательно, случаю точной разделимости. Если только L (или K) кратно периоду, то этот случай можно назвать левой (или правой) ортогональностью. Таким образом, если остаток детерминирован и S_N содержит периодическую компоненту, то мы наблюдаем два эффекта: специфическое поведение ошибок в случае длины окна, кратной периоду, и периодическое поведение ошибок в общем случае.

Ясно, что если остаток R_N содержит шум, то точная ортогональность не может быть достигнута. Рис. 3.12 демонстрирует, что уменьшение ошибок для специальных длин окон не происходит.

В случае комбинированных остатков (рис. 3.13), поведение ошибок проектора наследует свойства ошибок как для чисто случайных, так и для детерминированных остатков. Ниже мы покажем, что эта особенность справедлива и для других типов задач.

Чтобы показать, что тот факт, что шум красный (а не белый), не мешает извлечению сигнала, мы рассмотрим временной ряд (3.46) с красным шумом, см. рис. 3.13 (справа). Рис. 3.14 сравнивает MSD для различных структур остатков (напомним, что SNR одинаково). Видно, что красный шум дает немного худшую точность. Ошибки для временного ряда (3.45) лежат между ошибками для временного ряда (3.43) и временного ряда (3.44).

Подводя итог, в этом разделе мы продемонстрировали поведение ошибок оценок проектора для различных типов остатков. Однако нас обычно интересуют определенные особенности пространства сигналов, а не сам проектор. Поэтому результаты этого раздела дают лишь базовую информацию, которая может быть использована для объяснения дальнейших результатов.

3.4.3. Извлечение сигнала

Напомним, что восстановленный сигнал получается путем применения диагонального усреднения к восстановленной матрице, вычисленной по формуле $\tilde{\mathbf{S}} = \mathbf{U}_r \Lambda_r^{1/2} \mathbf{V}_r^T$, где $\mathbf{U}_r = [U_1 : \dots : U_r]$, $\mathbf{V}_r = [V_1 : \dots : V_r]$, и $\Lambda_r = \text{diag}(\lambda_1, \dots, \lambda_r)$; U_i , V_i и λ_i определены в (1.12). Заметим, что столбцы матрицы \mathbf{U}_r образуют базис для подпространства возмущенного сигнала для длины

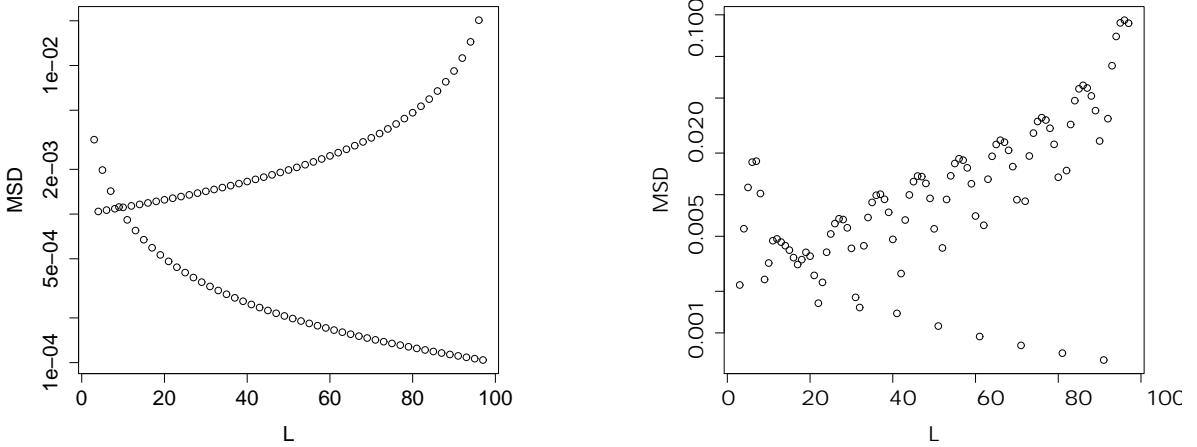


Рис. 3.11: MSD оценок проектора: детерминированные остатки (логарифмическая шкала); в.р. (3.42) (слева) и в.р. (3.43) (справа).

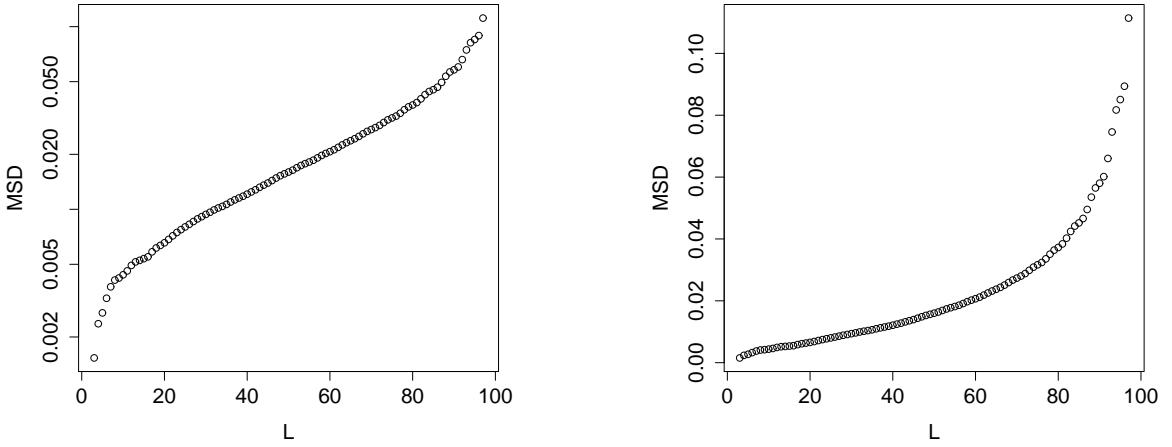


Рис. 3.12: MSD оценок проектора: детерминированные остатки, в.р. (3.44); логарифмический масштаб (слева) и оригинальный масштаб (справа).

окна L , в то время как столбцы матрицы \mathbf{V}_r формируют базис подпространства возмущенного сигнала для длины окна $N - L + 1$. Во-первых, это означает, что результаты одинаковы для длин окон L и $N - L + 1$. Затем, как показано на рис. 3.11–3.14, возмущение подпространства сигнала растет с увеличением L и уменьшается с увеличением $N - L + 1$. Таким образом, результирующие ошибки вызваны этими противоречивыми тенденциями. Рис. 3.12 (справа) демонстрирует, что скорость роста ошибок в оценке проектора больше при больших $L > N/2$. Поэтому нет ничего удивительного в том, что ошибки восстановления велики для малых длин окон из-за больших ошибок в \mathbf{V}_r . Однако вопрос об оптимальной длине окна остается открытым.

Далее мы рассмотрим зависимость ошибки восстановления $\tilde{s}_l - s_l$ от длины окна на нескольких примерах. Отметим, что явная асимптотическая форма известна для примера

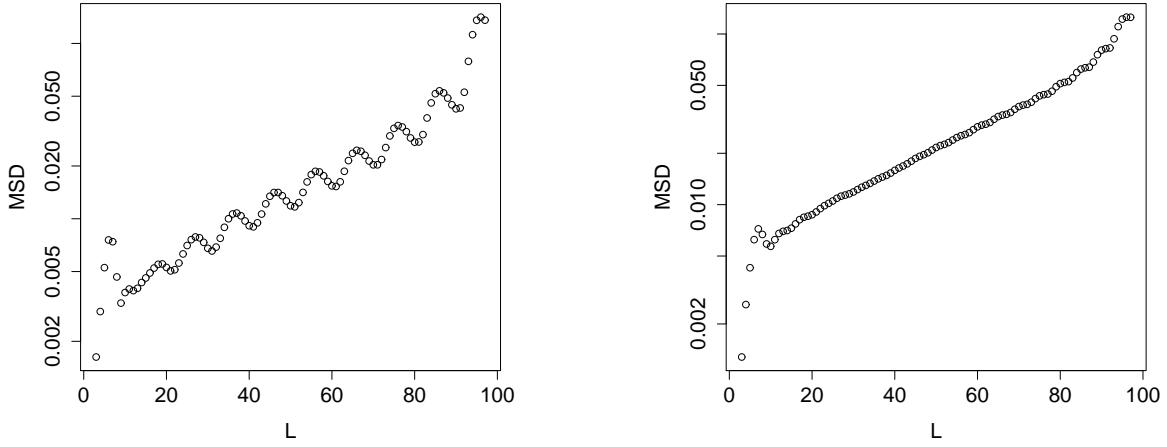


Рис. 3.13: MSD оценок проекторов: детерминированные остатки (логарифмическая шкала); смешанные остатки, в.р. (3.45) (слева) и остатки красный шум, в.р. (3.46) (справа)

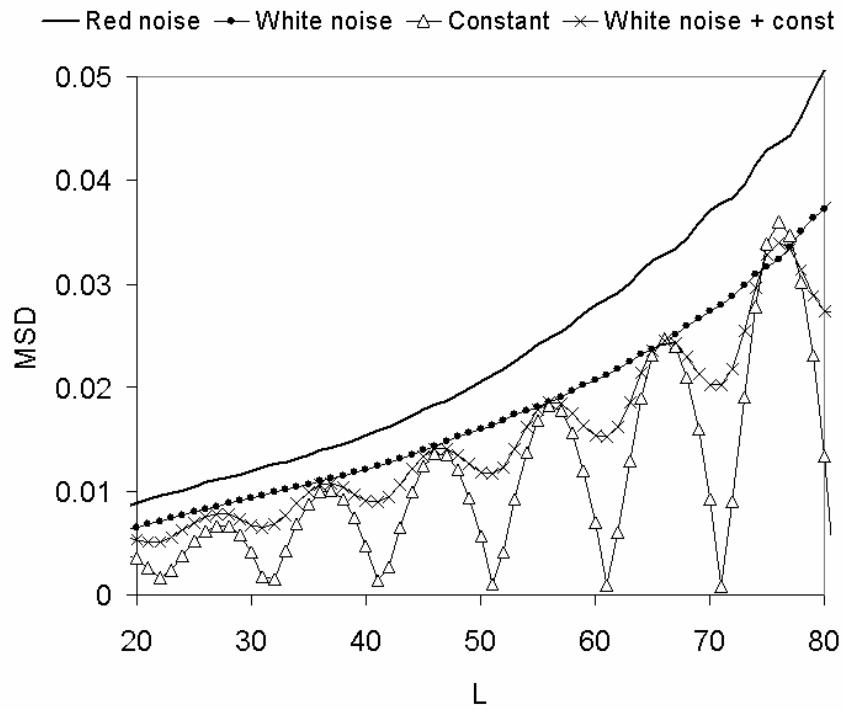


Рис. 3.14: MSD оценок проектора: различные типы остатков, в.р. (3.43)–(3.46)

зашумленного постоянного сигнала с $s_n \equiv c$ [38]. В статье [38] содержит явное выражение для дисперсии ошибок восстановления первого порядка, где первый порядок задается относительно величины возмущения сигнала рядом R_N , а не относительно N . Строго говоря, не обязательно, чтобы ошибка первого порядка была главным членом ошибки при $N \rightarrow \infty$. С помощью численного моделирования было проверено, что в случае чистого случайного шума мы можем рассматривать ошибку первого порядка как основной член ошибки (в общем случае это не так, см. [162]). Численное моделирование подтверждает, что качественные ре-

зультаты для постоянного сигнала справедливы для многих других типов сигналов, включая осцилляции. Для описания этих результатов приведем формулу для зависимости асимптотических ошибок от длины окна для постоянного сигнала [38].

Пусть длина окна $L \sim \beta N$, $0 < \beta \leq 1/2$, а l — индекс точки временного ряда, $l \sim \gamma N/2$, $0 \leq \gamma \leq 1$, при $N \rightarrow \infty$. Значение $\gamma = 1$ соответствует середине временного ряда; Следовательно, мы представляем формулу для первой половины временного ряда с длиной окна меньше, чем половина длины временного ряда. Тогда дисперсия ошибок первого порядка имеет следующий асимптотический вид:

$$Ds_l^{(1)} \sim \frac{\sigma^2}{N} \begin{cases} D_1(\beta, \gamma), & 0 \leq \gamma \leq 2\min(\beta, 1-2\beta), \\ D_2(\beta, \gamma), & 2\min(\beta, 1-2\beta) < \gamma < 2\beta, \\ D_3(\beta, \gamma), & 2\beta \leq \gamma \leq 1, \end{cases} \quad (3.47)$$

при $N \rightarrow \infty$, где

$$\begin{aligned} D_1(\beta, \gamma) &= \frac{1}{12\beta^2(1-\beta)^2} \left(\gamma^2(1+\beta) - 2\gamma\beta(1+\beta)^2 \right. \\ &\quad \left. + 4\beta(3-3\beta+2\beta^2) \right), \\ D_2(\beta, \gamma) &= \frac{1}{6\beta^2(1-\beta)^2\gamma^2} \left(\gamma^4 + 2\gamma^3(3\beta-2-3\beta^2) \right. \\ &\quad \left. + 2\gamma^2(3-9\beta+12\beta^2-4\beta^3) \right. \\ &\quad \left. + 4\gamma(-1+4\beta-3\beta^2-4\beta^3+4\beta^4) \right. \\ &\quad \left. + (8\beta-56\beta^2+144\beta^3-160\beta^4+64\beta^5) \right), \\ D_3(\beta, \gamma) &= \frac{2}{3\beta}. \end{aligned}$$

Точки изменения между случаями в (3.47) соответствуют $l = K - L$ (т.е. $\gamma = 2(1-2\beta)$) и $l = L$ ($\gamma = 2\beta$). Первая точка изменения присутствует, если $K < 2L$ ($\beta > 1/3$). Заметим, что формула (3.47) может быть распространена на длины окон $2 < L < N-1$ ($0 < \beta < 1$) и на индексы точек временного ряда $0 \leq l \leq N-1$ ($0 \leq \gamma \leq 2$) в силу симметрии ошибок относительно середины временного ряда и эквивалентности результатов при замене L на K ($\beta \leftrightarrow 1-\beta$).

Когда мы решаем задачу минимизации RMSE оценки s_l в фиксированной точке l , оптимальная длина окна изменяется от $N/3$ до $N/2$, см. [38]. Это означает, что даже в случае с постоянным сигналом оптимальная длина окна, которая минимизирует ошибки восстановления в целом, зависит от важности (веса) каждой точки временного ряда. В любом случае, общая рекомендация заключается в следующем: выбирать длину окна чуть меньше половины длины временного ряда N . Обратите внимание, что оптимальная длина окна обеспечивает значительное улучшение уровня ошибок (по сравнению с выбором $L = N/2$) в крайних точках временного ряда, то есть при $l/N \approx 0$.

Для проекторов в разделе 3.4.2 было показано, что для зашумленного синусоидального сигнала (т.е. если остатки не содержат детерминированной составляющей), кратность длины окна периоду синусоиды не является важным вопросом. Наличие детерминированного компонента в остатке делает эту делимость важной. Аналогичный эффект имеет место для ошибок восстановления $\|\tilde{S}_N - S_N\|$, см. рис. 3.15 для временного ряда (3.43)–(3.46) с теми же параметрами.

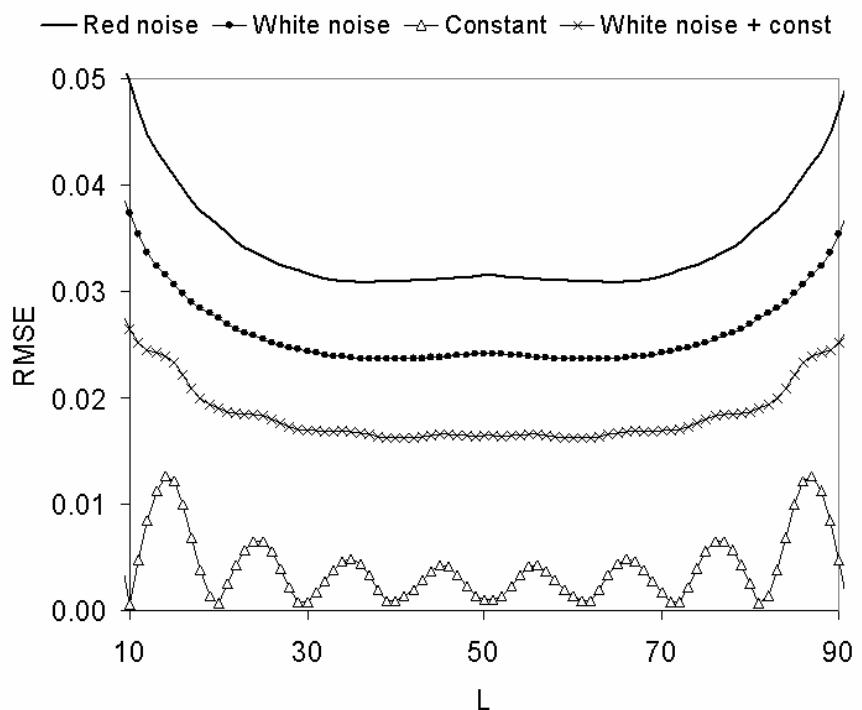


Рис. 3.15: RMSE оценок сигнала: различные типы остатков, в.р. (3.43)–(3.46)

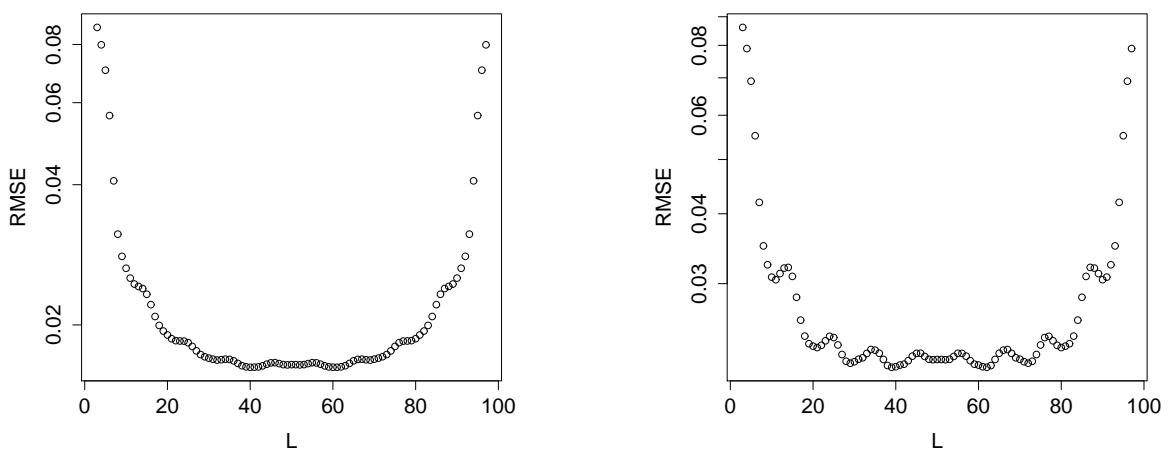


Рис. 3.16: RMSE оценок сигнала: смешанные остатки, в.р. (3.45); весь ряд (слева) и последние 10 точек (справа)

Чтобы изучить влияние длины окна на RMSE, мы рассмотрим RMSE для восстановления десяти последних точек сигнала. Сравнение графиков на рис. 3.16 показывает, что влияние кратности длины окна к периоду синусоиды сильнее для крайних точек.

На рис. 3.15 мы видим, что оптимальная длина окна близка к $0.4N$ в случае случайного остатка. Однако делимость длины окна на период (если остатки содержат детерминированный компонент) может быть более важной чем корректировка, например, чем переход от $N/2$ к $0.4N$.

3.4.4. Рекуррентный прогноз SSA

Зависимость от длины окна

Рассмотрим зависимость точности прогноза от длины окна L (ранг сигнала r считаем заданным заранее). Отметим, что в процедуре прогнозирования используются два объекта: само ЛРС и начальные данные для этого ЛРС, взятые из последних точек восстановленного сигнала \tilde{S}_N . Обозначим вектор, построенный из $L - 1$ последних точек сигнала, через V и вектор, построенный из $L - 1$ последних восстановленных (т.е. возмущенных) точек сигнала, обозначим $V + \Delta V$. Аналогично, обозначим вектор коэффициентов истинной минимальной нормы ЛРС через A и вектор оцененных коэффициентов ЛРС через $A + \Delta A$. Тогда ошибка прогноза составляет $A^T \Delta V + (\Delta A)^T V + (\Delta A)^T \Delta V$.

Таким образом, ошибка первого порядка состоит из двух видов ошибок:

1. ошибки в коэффициентах ЛРС, которые вызваны ошибкой проекции на подпространство сигнала $(\Delta A)^T V$;
2. ошибки восстановления сигнала $A^T \Delta V$.

Давайте исследуем эти два источника ошибок отдельно. Для этого применим ЛРС, оцененное с помощью окна с длиной L_{LRR} , к истинным значениям сигнала и применим истинное ЛРС к оцененным значениям сигнала, которые были восстановлены с длиной окна L_{rec} .

Мы рассматриваем временной ряд (3.44) с $\sigma = 0.1$, $N = 399$ и $\ln b = 0, 0.01, -0.01$. Оценка (по 1000 промоделированным реализациям) RMSE прогноза на один шаг вперед изображена на рис. 3.17 для $L_{rec} = 200$. Значения L_{LRR} изменяются от 20 до 380 с шагом 20. Аналогичные графики для различных L_{rec} представлены на рис. 3.18. Обозначения оси x соответствуют значениям L_{rec} .

Линия, обозначенная ‘total’, показывает точность (RMSE) прогноза с соответствующей длиной окна. Линия, обозначенная ‘LRR’(ЛРС), соответствует $(\Delta A)^T V$. Ошибки ‘LRR’ выглядят как ошибки оценки P_r проектора на пространство сигналов (см. рис. 3.12 (справа)). Это неудивительно, поскольку коэффициенты ЛРС пропорциональны проекции $\mathbf{U}_{r+1,L} \mathbf{U}_{r+1,L}^T \mathbf{e}_L = P_r^\perp \mathbf{e}_L$. Естественно, ошибки ‘LRR’ не зависят от длины окна L_{rec} , используемого для восстановления. Типичное поведение этой части ошибки таково: чем больше длина окна, тем большая ошибка.

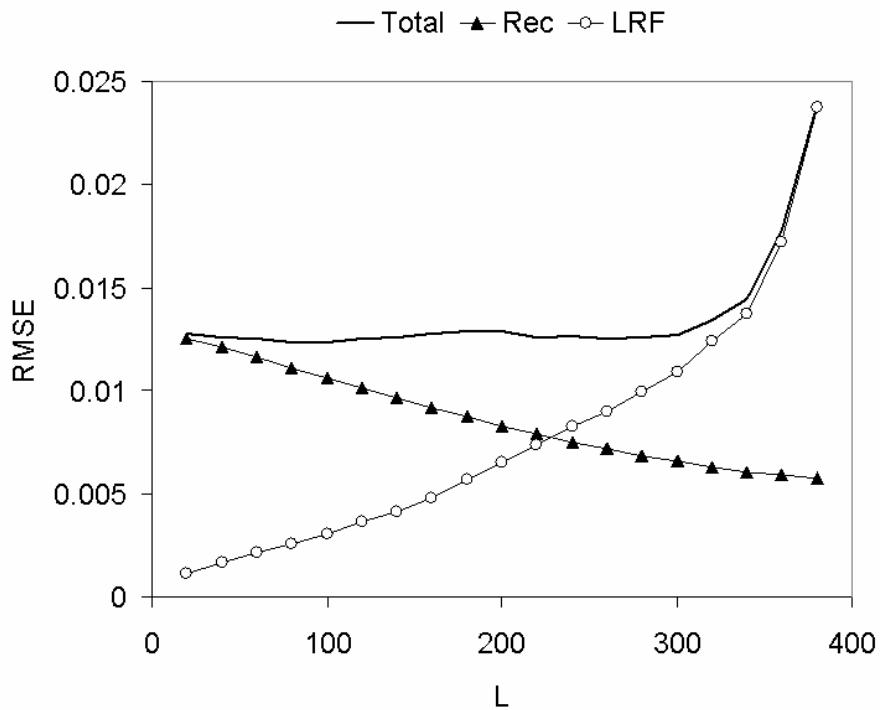


Рис. 3.17: RMSE прогноза как функция L_{LRR} : в.р. (3.44) с $b = 1$, $L_{\text{rec}} = 200$

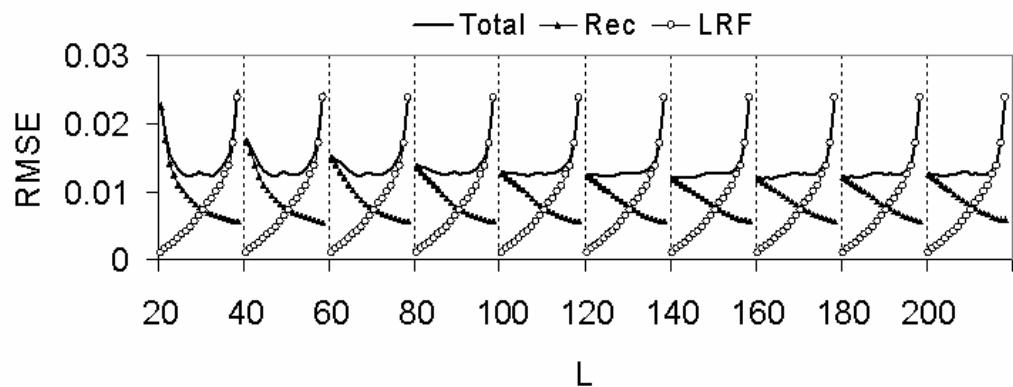


Рис. 3.18: RMSE прогноза: в.р. (3.44) с $b = 1$, различные L_{rec}

Линия, обозначенная ‘Rec’, соответствует $A^T \Delta V$. Видно, что чем больше длина окна, тем меньше ошибка. Это можно интерпретировать следующим образом: посторонние корни, расположенные близко к равномерному распределению на окружности, компенсируют друг друга (см. [112, 107] для нескольких результатов такого рода).

Рис. 3.17–3.18 показывают, что точность прогнозов стабильна в широком диапазоне длин окон. В частности, L_{LRR} и L_{rec} немного меньше $N/2$ являются вполне подходящими. Также можно взять либо небольшие L_{LRR} и $L_{\text{rec}} \sim N/2$, либо $L_{\text{LRR}} \sim N/2$ и маленькое L_{rec} . Первый вариант является предпочтительным, так как ошибки меньше и более устойчивы к изменениям длины окна, см. рис. 3.18.

Рекомендации по выбору длины окна, естественно, зависят от форм сигнала и остатка. Рис. 3.19 и 3.20 содержат RMSE для затухающих синусоид. Интерпретация рисунков и поведение RMSE аналогичны.

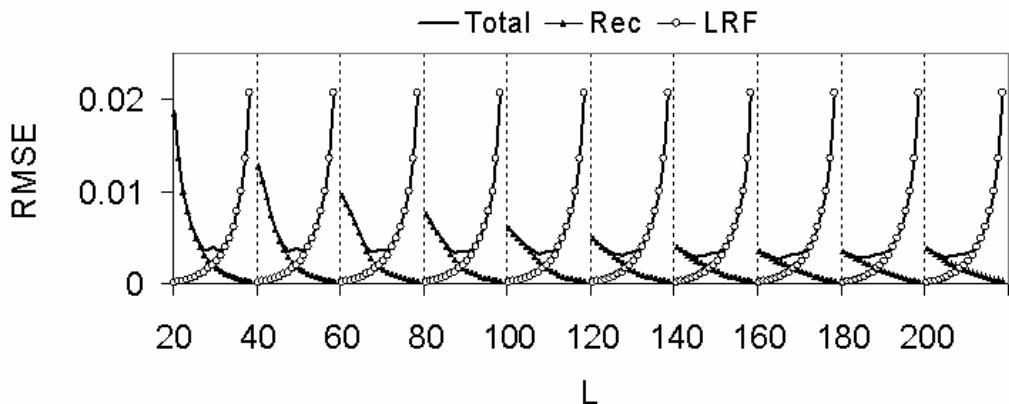


Рис. 3.19: RMSE прогноза: в.р. (3.44) с $b < 1$,
разные L_{rec}

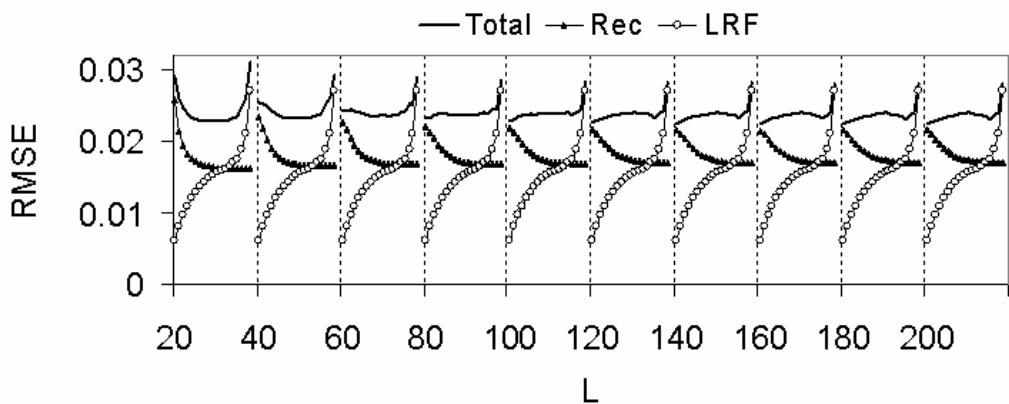


Рис. 3.20: RMSE прогноза: в.р. (3.44) с $b > 1$,
разные L_{rec}

Детерминированный остаток может обеспечить специфическое поведение ошибок. В частности, для временного ряда (3.42) мы получаем, что выбор четного N и нечетных значений L_{LRR} обеспечивает уменьшение ошибок проекторов при увеличении длины окна (см. рис. 3.11 (слева)). Таким образом, при таком выборе L и N мы наблюдаем аналогичное поведение для двух источников ошибок прогноза и оптимальным выбором является как можно большая длина окна.

3.4.5. Подпространственные методы оценки параметров

Рассмотрим выбор параметров SSA для задачи оценки параметров временного ряда, таких как частота временного ряда.

В статье [164] содержится следующий теоретический результат: при оценке частоты зашумленной синусоиды, асимптотическая ($N \rightarrow \infty$) дисперсия ошибки первого порядка имеет порядок $1/(K^2 L)$ и выбор длины окна симметричен относительно $N/2$. Поэтому асимптотическая оптимальная длина окна равна $N/3$ или $2N/3$. Численные эксперименты подтверждают этот вывод. Заметим, что не всегда верно, что первый порядок (по отношению к уровню возмущения) ошибки является ошибкой главного порядка, поскольку длина временного ряда стремится к бесконечности. Поэтому лучше проверить соответствие между ошибкой первого порядка и общей ошибкой с помощью моделирования.

В [165] явная форма асимптотической дисперсии ошибки первого порядка выведена в общем случае затухающих комплексных экспонент. В случае незатухающих комплексных экспоненциалов, полученная форма совпадает с формой в [164]. Что касается затухающих комплексных экспонент, результат заключается в том, что оптимальная длина окна лежит между $N/3$ и $N/2$ и приближается к $N/2$ по мере увеличения коэффициента затухания. В [165] показано, что для $s_n = \exp((\alpha + i\beta)n)$, $i = \sqrt{-1}$, дисперсии первого порядка оценок ESPRIT α и β равны. Поэтому оптимальные длины окон одинаковы для оценок коэффициента затухания α и частоты β .

В предыдущих разделах мы показали, что отделимость сигнала от детерминированных и стохастических остатков имеет разную природу и, следовательно, приводит к различным последствиям. Рассмотрим, как это различие проявляется в проблеме оценки частоты с помощью ESPRIT. Проведем моделирование для временного ряда (3.43)–(3.46) с $c = \sigma = 0.1$, $b = 1$, $N = 100$.

Рис. 3.21 содержит результаты для детерминированного возмущения, включая специфическое поведение RMSE в случае односторонней (левой) ортогональности. Рис. 3.22 содержит RMSE оценок частоты для различных видов остатков. Поведение ошибок очень похоже на поведение ошибок при восстановлении сигнала, см. рис. 3.15. Также ошибки оценок частоты и экспоненциальной скорости (коэффициента затухания) примерно равны, если L пропорционально N . Основное отличие от ошибок восстановления заключается в размере ошибок, который значительно меньше. Поэтому для оценки RMSE с достаточной точностью мы используем 1000 вместо 100 имитационных рядов.

Рис. 3.23 обращает внимание на поведение ошибки для малых длин окон. Видно, что доминирование ошибок для временного ряда (3.43) с детерминированными остатками является отличительной особенностью.

3.4.6. Скорость сходимости

В статье [162] содержатся теоретические результаты о сходимости (когда длина временного ряда N стремится к бесконечности) для методов, которые основанных на оценке подпространства сигнала. Здесь мы исследуем скорость сходимости на примерах.

Рассмотрим два временных ряда с длинами N_1 и N_2 такими, что $N_2 = 4N_1$. Пусть RMSE

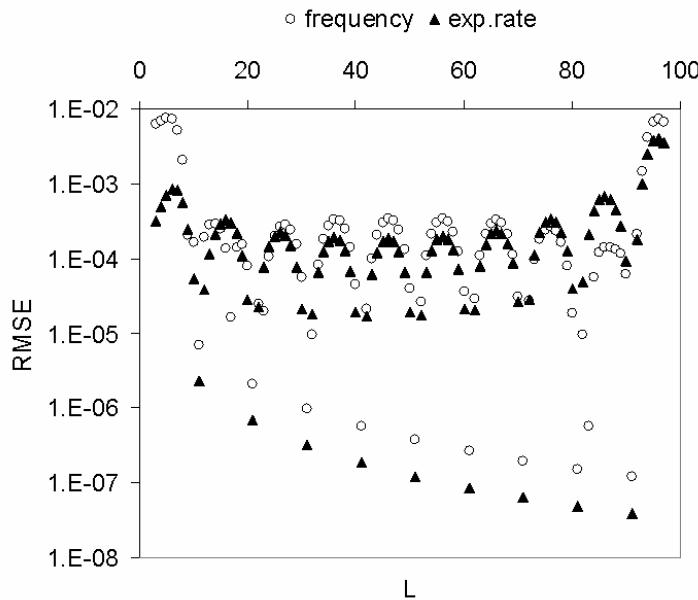


Рис. 3.21: RMSE оценок частоты и эксп.скорости: в.р. (3.43) (логарифмическая шкала)

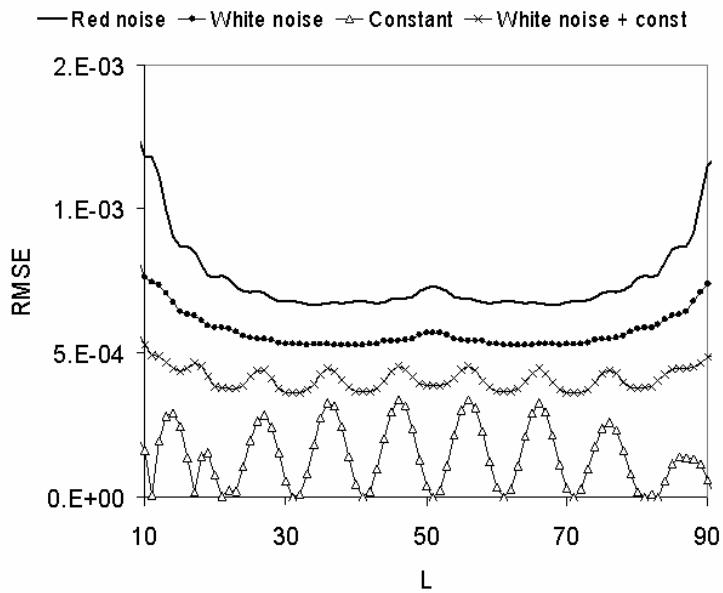


Рис. 3.22: RMSE оценок частоты: различные типы остатков, в.р. (3.43)–(3.46), $L \sim N/2$

является мерой точности. Если остаток R_N является случайным, то для оценки RMSE проводим имитационное моделирование. Обозначим отношение RMSE для длин окон N_1 и N_2 через $\Delta = \text{RMSE}_1/\text{RMSE}_2$. Тогда $\Delta = 8$ означает скорость сходимости $1/N^{1.5}$, $\Delta = 2$ соответствует скорости сходимости $1/N^{0.5}$, а $\Delta = 1$ означает, что сходимости нет вообще. Чтобы оценить Δ , мы используем $N_1 = 6399$ и $N_2 = 25599$ (мы выбрали нечетные длины временных рядов, чтобы рассматривать $(N+1)/2$ как одну из длин окна).

Мы рассматриваем примеры со следующими тремя видами возмущения сигнала: константой, шумом и суммой шума и константы. Также мы рассматриваем два вида случайного шума: белый и красный. Мы проводим численные эксперименты для временного ряда

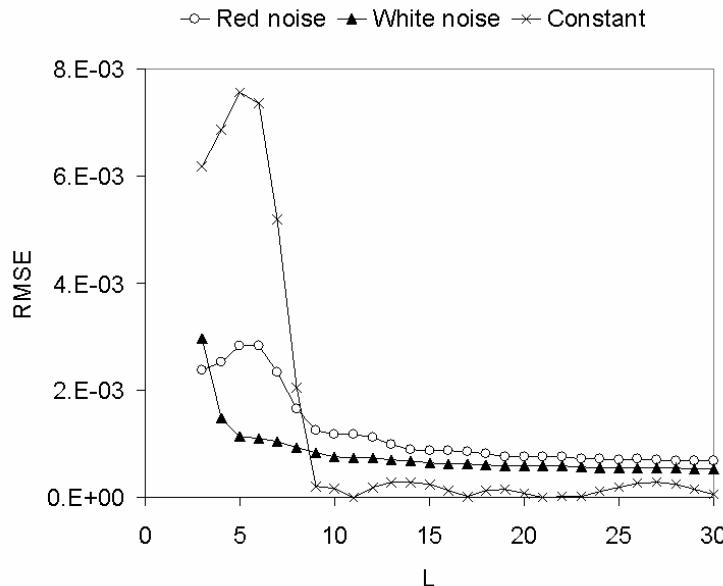


Рис. 3.23: RMSE оценок частоты: различные типы остатков, в.р. (3.43)–(3.46), L мал

Таблица 3.3: Скорость сходимости восстановления: оценка Δ

L	c	c+wn	wn	rn
$r+1$	1.0	1.0	1.0	1.0
20	—	1.0	1.0	1.0
25	1.0	1.0	1.0	1.0
$(N+1)/2 - 5$	8.0	1.9	2.0	2.1
$(N+1)/2$	—	1.9	2.0	2.1

Таблица 3.4: Скорость сходимости оценок проектора: оценка Δ

L	c	c+wn	wn	rn
$r+1$	1.0	1.0	2.2	1.0
20	—	3.1	2.8	1.1
25	1.0	1.0	3.0	1.1
$(N+1)/2 - 5$	4.0	2.0	2.0	2.0
$(N+1)/2$	—	2.0	2.0	2.0

(3.43)–(3.46) с $\sigma = 0.1$, $b = 1$, $\alpha = 0.5$. Далее мы рассчитаем оценки частоты и экспоненциальной базы, используя метод LS-ESPRIT (разница с результатами TLS-ESPRIT невелика и не влияет на выводы).

Таблицы 3.3–3.6 включают результаты сходимости на основе 1000 симуляций. Столбец ‘c’ соответствует временному ряду (3.43) с постоянными остатками, столбцы ‘wn’ и ‘rn’ содержат результаты для белого шума (временной ряд (3.44)) и красного шума (временной ряд (3.46)) соответственно, а столбец ‘c+wn’ включает оценки Δ для временного ряда (3.45) с комбинированным возмущением.

Кажется разумным, что скорости сходимости для фиксированной длины окна и для длины окна, пропорциональной N , отличаются. Кроме того, может иметь значение кратность длины окна периоду синусоидального сигнала (10 в рассматриваемых примерах). Поэтому мы анализируем два набора длин окон. Первый набор включает фиксированные длины окон: минимальная $L = r + 1$, где $r = 2$ — ранг синусоиды, $L = 20$ — кратная 10, а $L = 25$ — общий случай. Второе множество содержит две длины окна, близкие к $N/2$: $L = (N+1)/2$ кратно 10 и $L = (N+1)/2 - 5$ — общий случай. Заметим, что если существует точная разделимость (что возможно только в случае детерминированных остатков), то отношение Δ не может быть

Таблица 3.5: Скорость сходимости оценок экспоненциальной базы: оценка Δ

L	c	c+wn	wn	rn
$r+1$	—	4.1	4.1	4.0
20	—	4.0	4.1	3.9
25	1.0	2.8	4.1	3.9
$(N+1)/2 - 5$	16.0	8.2	8.2	8.1
$(N+1)/2$	—	8.2	8.2	8.1

Таблица 3.6: Скорость сходимости оценок частоты: оценка Δ

L	c	c+wn	wn	rn
$r+1$	1.0	1.0	2.0	1.0
20	—	2.7	2.5	1.0
25	1.0	1.0	2.7	1.0
$(N+1)/2 - 5$	16.0	8.2	8.4	8.2
$(N+1)/2$	—	8.2	8.4	8.2

вычислено (знак ‘—’ в таблицах). Мы не рассматриваем окна длиной $L > (N+1)/2$, так как эти значения L приводят либо к той же, либо к худшей скорости сходимости по сравнению с длиной окна $L \leq (N+1)/2$.

Обсудим результаты, представленные в таблицах 3.3–3.6 для длины окна, стремящейся к бесконечности, и для фиксированной длины окна по отдельности.

Длина окна $L \sim N/2$ Моделирование дает стабильные оценки скорости сходимости для длины окна L , равной половине длины временного ряда (и, в более общем случае, для длин окон, пропорциональных N):

(A) “сигнал+шум” (временной ряд (3.44), (3.46))

- (a) скорость сходимости проектора на сигнальное подпространство составляет $1/N^{0.5}$,
- (b) скорость сходимости восстановления всего сигнала (средняя ошибка) составляет $1/N^{0.5}$,
- (c) скорость сходимости оценок частоты и экспоненциальной скорости составляет $1/N^{1.5}$;

(B) “сигнал+константа” (временной ряд (3.43))

- (a) скорость сходимости проектора на подпространство сигнала почти равна $1/N$,
- (b) скорость сходимости восстановления всего сигнала (средняя ошибка) составляет почти $1/N^{1.5}$,
- (c) скорость сходимости оценок частоты и экспоненциальной скорости составляет почти $1/N^2$.

Теоретические результаты об ошибках восстановления в частном случае зашумленного константного сигнала [38], см. (3.47), обеспечивают поддержку части выводов, полученных в результате моделирования. Результаты по RMSE для оценок частоты и экспоненциального основания подтверждаются в [164], рассматривается случай зашумленной синусоиды. Отметим, что скорость сходимости $1/N^{1.5}$ не удивительна, так как нижняя граница Рао-Крамера для дисперсии оценок частоты имеет тот же порядок (см., например, [166, 167]). С другой стороны, нижняя граница Рао-Крамера для дисперсии оценок амплитуды синусоиды имеет порядок $1/N^{0.5}$, что соответствует скорости сходимости восстановленного сигнала. Моделирование для временного ряда (3.46) подтверждает, что для $L \sim N/2$ остатки с красным шумом обеспечивают такую же скорость сходимости что и остатки с белым шумом.

Видно, что скорость сходимости в примерах с чисто случайными остатками намного хуже, чем в примерах с детерминированными (постоянными) остатками. Как и следовало

ожидать, пример с комбинированными остатками наследует худший случай. Моделирование подтверждает что в случае постоянного остатка, смешанного со стохастическим компонентом (временной ряд (3.45)) и длиной окна L , пропорциональной N , скорость сходимости такая же, как и для случая (A).

Фиксированная длина окна $L = L_0$ Рассмотрим случай фиксированной длины окна L и $N \rightarrow \infty$. Поведение скорости сходимости более сложное, чем описанное выше.

Анализ *ошибки восстановления* показывает следующее поведение:

- (A) для “сигнал+шум” (временной ряд (3.44), (3.46)), нет сходимости к сигналу, даже при длине окна, кратной периодам сигнала (или шума) (в рассматриваемом примере кратной 10);
- (B) для “сигнал+константа” (временной ряд (3.43)), сходимость имеет место, только если L (или $K = N - L + 1$) кратно периоду; в общем случае сходимость отсутствует.

Следовательно, если остатки содержат шум, то сходимость отсутствует. Таким образом, в общем случае, малая длина окна не подходит для задач восстановления сигнала.

Теперь рассмотрим *ошибки проектора и оценок параметров*. Отличие от поведения ошибок восстановления состоит в наличии сходимости для малых L и остатков с белым шумом. Однако моделирование для временного ряда (3.46) с остатками с красным шумом демонстрирует отсутствие сходимости для проекции и оценки частоты. В таблицах 3.4 и 3.6, отсутствие сходимости соответствует значениям ‘1.0’ и ‘1.1’ в колонке под названием ‘тп’. Сходимость оценок экспоненциальной базы все же имеет место, см. таблицу 3.5.

Вопрос в том, как детерминированный (константная, в наших примерах) составляющая остатков в виде белого шума влияет на сходимость для малых L . К сожалению, ошибки для проектора и оценок параметров сходятся к 0 для $L = L_0$ только если длина окна кратна периодам сигнала.

Таким образом, можно сделать вывод, что использование малых длин окон для оценки частоты возможно только если остатки являются чистым белым шумом, то есть не содержат детерминированных компонентов и являются независимыми. В противном случае сходимости не будет.

Сравним ошибку оценки для остатков с белым и красным шумом независимо от отсутствия сходимости в последнем случае. На рис. 3.24 показаны абсолютные значения ошибок оценки для $L = 10$. Первые символы названий линий означают типы оцениваемых объектов: проектор (‘proj’), экспоненциальная база (‘base’), или частота (‘freq’); последние символы обозначают типы остатков. Видно, что размеры ошибок для остатков красного и белого шума сопоставимы для рассматриваемых длин временных рядов. Этот эффект достаточно устойчив для различных параметров модели временного ряда. Таким образом, мы можем проводить оценку с хорошей точностью даже в случае отсутствия сходимости. Это можно объяснить результатами [162, формула (2.15)], где получены верхние границы для ошибок

проектора. Главный член верхней границы (это константа, зависящая от частоты синусоиды, параметров красного шума и длины окна) дает правильный порядок ошибок. Более того, [162, формула (2.9)] дает примерно такую же малую величину $1.3 \cdot 10^{-3}$, как представлено на рис. 3.24, линия помечена ‘proj_rn’. Расчеты подтверждают, что эта величина примерно равна $K(\mathbf{S}\mathbf{S}^T)^{\dagger} \boldsymbol{\Sigma} (\mathbf{I} - \mathbf{U}_r^{(s)}(\mathbf{U}_r^{(s)})^T)$, где $\boldsymbol{\Sigma}$ — автоковариационная $L \times L$ матрица рассматриваемого красного шума. Обратите внимание, что этот член не сходится к 0, когда длина временного ряда стремится к бесконечности.

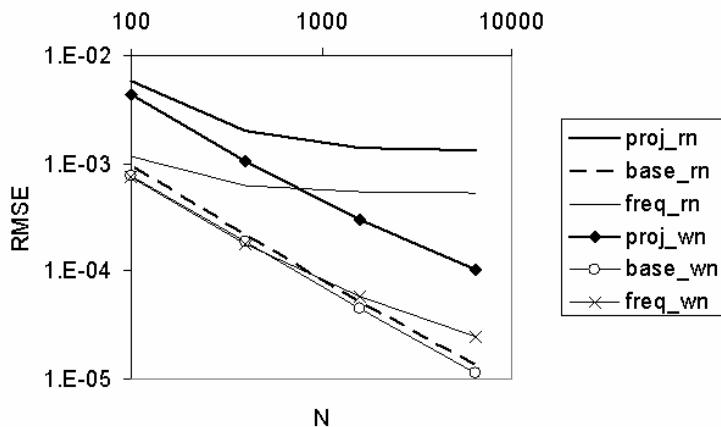


Рис. 3.24: Сравнение RMSE для остатков красного и белого шума

3.4.7. Выбор длины окна и разделимость

Модулированная синусоида

Обратим внимание, что способность SSA-подобных методов извлекать экспоненциально модулированные (затухающие) синусоиды подразумевает, что метод SSA не является просто спектральным методом, даже если мы не применяем SSA для выделения тренда. Заметим, что как затухающие, так и незатухающие синусоиды имеют SSA-ранг 2 (или ранг 1, если синусоиды рассматриваются в комплексно-значном случае). Эта особенность SSA значительно расширяет набор временных рядов, которые подходят для анализа SSA. Большинство классических методов (например, анализ Фурье, сезонное разложение) имеют дело либо с постоянными амплитудами, либо с пропорциональными трендом и могут быть сведены к постоянным амплитудам путем перевода временных рядов в логарифмическую шкалу. Однако временные ряды, состоящие из нескольких затухающих синусоид, не могут быть сведены к периодическим временным рядам или к мультиплекативной периодичности. Рассмотрим, например, сезонный компонент, содержащий как годовые, так и квартальные колебания, и пусть амплитуда годовой периодичности увеличивается, в то время как амплитуда квартальных колебаний уменьшается. Тогда многие классические методы не сработают, в то время как SSA-подобные методы могут легко извлечь такую сезонность.

Если поведение модуляций сложнее экспоненциального, то SSA-подобные методы могут столкнуться с трудностями. Эти методы все же могут извлечь такие колебания, однако возникает вопрос о правильном выборе длины окна.

Давайте сформулируем этот вопрос в более конкретной форме. Рассмотрим сигнал в виде $s_n = A(n) \cos(2\pi n \omega)$, где $A(n)$ — медленно (по сравнению с ω) изменяющаяся функция. Вопрос: есть ли примеры, когда выбор длины окна близкой к $N/2$ не является хорошим.

Мы рассматриваем временной ряд $x_n = s_n + r_n$ с

$$s_n = \cos(2\pi n / 19) + \cos(2\pi n / 21), \quad r_n = \varepsilon_n. \quad (3.48)$$

Здесь $s_n = A(n) \cos(2\pi n / 20)$ с $A(n) = 2 \cos(\pi n (1/19 - 1/21))$ является модулированной синусоидой с частотой $1/20$. Сигнал имеет ранг 4 и асимптотически отделим от шума, постоянного остатка и других. У нас есть две альтернативные возможности. Первая возможность — взять L близко к $N/2$ (например, между $N/3$ и $N/2$) и извлечь сигнал по четырем ведущим собственным тройкам. Вторая альтернатива заключается в том, чтобы взять длину окна L настолько малой, что амплитуда сигнала почти постоянна в пределах подрядов длины L , а затем выделить сигнал по двум ведущим компонентам. В последнем случае левые сингулярные векторы близки к незатухающим синусоидам, а модуляцию улавливают правые сингулярные векторы.

Численное моделирование (см. рис. 3.25 для длин временных рядов равных 99, 199, 399 и 999) показывают, что нет четкого выбора между описанными альтернативами. Если длина временного ряда достаточно велика для приближенной разделимости, то лучше выбрать $L \sim N/2$. В противном случае, длина окна, близкая к паре периодов, обеспечивает лучшую точность. Недостатком последнего варианта является то, что обычно мы не знаем периода и поэтому мы не можем угадать правильную длину окна.

Таким образом, при условиях приближенной разделимости (можно проверить, выполнены ли эти условия с помощью анализа результатов разложения, см. [1]) наилучшее восстановление сигнала использует длину окна, близкую к $N/2$ и число соответствующих собственных троек равно рангу сигнала. Преимуществами такого выбора являются лучшая точность и независимость выбора длины окна от неизвестных значений периодов. В рассматриваемом примере для $L = N/2$ восстановление по четырем собственным тройкам хорошо работает, начиная с $N = 160$. Однако, выбор $L = N/2$ подходит даже для $N < 120$, если мы производим восстановление по двум собственным тройкам. Таким образом, существует ограниченный диапазон длин временного ряда (примерно от 120 до 160), где выбор длины окна, близкой к $N/2$, не может обеспечить адекватную точность восстановления сигнала.

Таблица 3.7 содержит RMSE для восстановления сигнала с использованием двух длин окна: $L = 40$ (два периода) и $L = N/2$. Значения, выделенные жирным шрифтом, указывают на меньшие ошибки.

Случай модуляции сложной формы Рассмотрим случай, когда модулированный сигнал

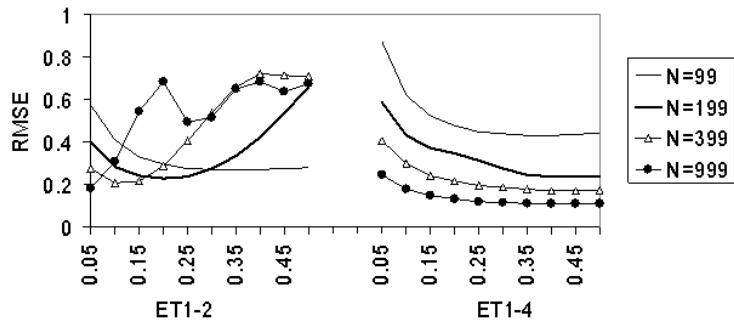


Рис. 3.25: RMSE восстановления: зависимость от L/N для в.р. (3.48), ET1–2 и ET1–4

Таблица 3.7: RMSE восстановление: различные варианты параметров для в.р. (3.48)

N	$L = 40$	$L = N/2$	$L = N/2$
	2ET	2ET	4ET
99	0.27	0.27	0.45
159	0.23	0.42	0.24
199	0.22	0.65	0.25
399	0.20	0.70	0.16
999	0.19	0.68	0.11

не является сигналом конечного ранга. Пусть $N = 399$ и $x_n = s_n + r_n$, где

$$s_n = A(n) \cos(2\pi n/20), \quad r_n = \sigma \varepsilon_n, \quad (3.49)$$

$A(n) = \cos(2\pi n^2/10^5)$. На рис. 3.26 и 3.28 показаны исходные временные ряды с различными уровнями шума, а рис. 3.27 и 3.29 содержат ошибки восстановления по 2, 4, 6 и 8 ведущим собственным тройкам. Выбор длины окна для восстановления сигнала не имеет решающего значения для низких уровней шума, так как мы можем достичь хорошей точности правильно выбирая собственные тройки для восстановления (чем больше длина окна $L \leq N/2$, тем большее количество собственных троек выбирается для восстановления).

Рис. 3.29 показывает, что для высокого уровня шума выбор длины окна, близкой к $N/2$, не обеспечивает наилучшую точность. В частности, использование $L = 200$ с восстановлением по четырем ведущим собственным тройкам немного хуже, чем использование $L = 40$ (два основных периода) и двух ведущих собственных троек. Однако, если взять, скажем, $L = 80$, то восстановление по двум ведущим собственным тройкам будет менее точным. Таким образом, выбор $L \sim N/2$ является вполне подходящим и может быть улучшен только с помощью дополнительной информации (например, значения периода) об анализируемом временном ряде.

Отметим, что в этом подразделе мы рассматривали выбор длины окна для восстановления сигнала. Оценка основной частоты (например, с помощью ESPRIT) это отдельная задача, требующая специальных постановок и подходов к ее решению.

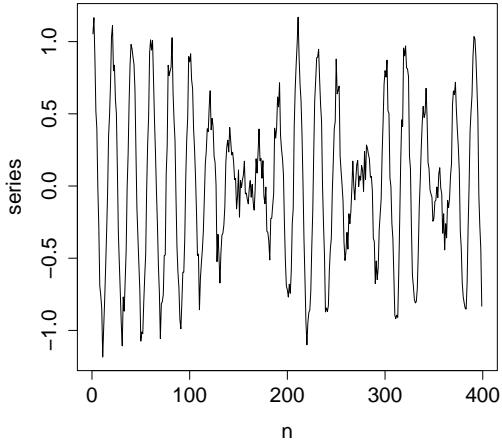


Рис. 3.26: Исходный временнй ряд:
в.р. (3.49), $\sigma = 0.1$

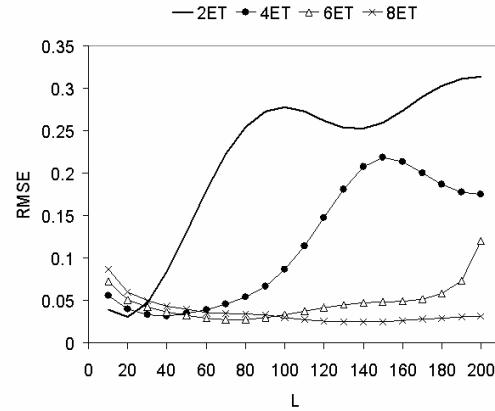


Рис. 3.27: RMSE восстановления:
в.р. (3.49), $\sigma = 0.1$

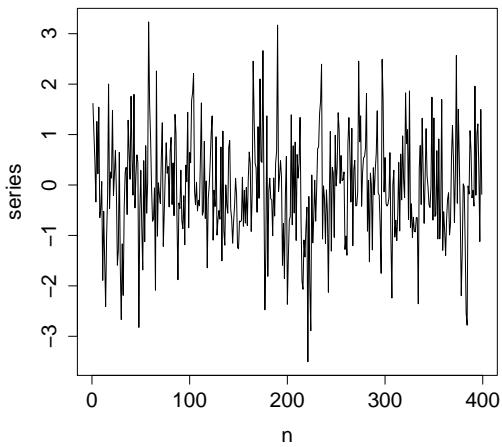


Рис. 3.28: Исходный временнй ряд:
в.р. (3.49), $\sigma = 1$

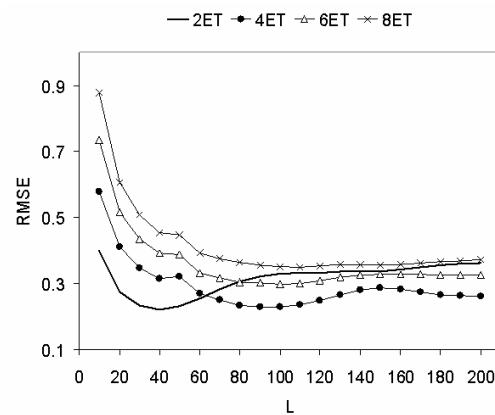


Рис. 3.29: RMSE восстановления:
в.р. (3.49), $\sigma = 1$

Проблема смещивания

Одной из проблем, с которой сталкивается SSA, является возможное отсутствие сильной разделимости [1] в условиях слабой разделимости. Эта проблема вызвана равенством сингулярных значений в SVD траекторных матриц сигнала и остатка.

Рассмотрим в качестве сигнала некоторую детерминированную медленно меняющуюся компоненту (т.е. тренд). Сигнал не обязательно имеет конечный ранг. Более вероятно, что он может быть аппроксимирован временным рядом конечного ранга r_{appr} . Пусть Q — доля r_{appr} квадратов ведущих сингулярных значений в SVD траекторной матрицы тренда. Обозначим через $\lambda_{\max}^{\text{resid}}$ максимальное сингулярное значение, порожденное траекторной матрицей остат-

ка, и $\lambda_{\min}^{\text{trend}} r_{\text{appr}}$ -е сингулярное значение траекторной матрицы тренда.

Случай приближенной разделимости соответствует большому Q при условии $\lambda_{\max}^{\text{resid}} < \lambda_{\min}^{\text{trend}}$. Если Q фиксировано, то, в общем случае, чем больше длина окна, тем больше ранг аппроксимирующего временного ряда и тем меньше $\lambda_{\min}^{\text{trend}}$. Это наблюдение может привести к тому, что оптимальная длина окна будет значительно меньше $N/2$.

Мы рассмотрим пример

$$s_n = \cos(2\pi n^2/10^5), \quad r_n = \varepsilon_n + 0.5 \cos(2\pi n/10), \quad (3.50)$$

$N = 199$ (рис. 3.30). Легко проверить численно, что для $L = 100$ мы можем получить $r_{\text{appr}} = 2$ при $Q = 99.5\% = 92\% + 7.5\%$ (сингулярные значения равны 74 и 21), тогда как для $L = 30$, $r_{\text{appr}} = 1$ с $Q = 98\%$ (сингулярное значение равно 41). Однако, остаток дает максимальное сингулярное значение, равное примерно 25 для $L = 100$ и 15 для $L = 30$. Это означает, что для $L = 100$ с $Q = 99.5\%$ мы не имеем сильной разделимости. Таким образом, мы можем достичь $Q = 92.5\%$ для $L = 100$ и $Q = 98\%$ для $L = 30$ (конечно, это только примерно) чтобы удовлетворить $\lambda_{\max}^{\text{resid}} < \lambda_{\min}^{\text{trend}}$, что подтверждает рис. 3.31. Заметим, что после извлечения тренда с $L_1 = 30$ периодичность может быть извлечена из остатка с длиной окна, близкой к $L_2 = 100$. В [1] этот метод называется последовательным SSA.

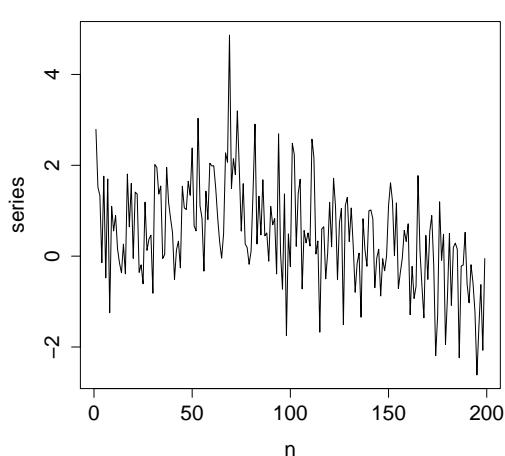


Рис. 3.30: Начальный временной ряд:
в.р. (3.50)

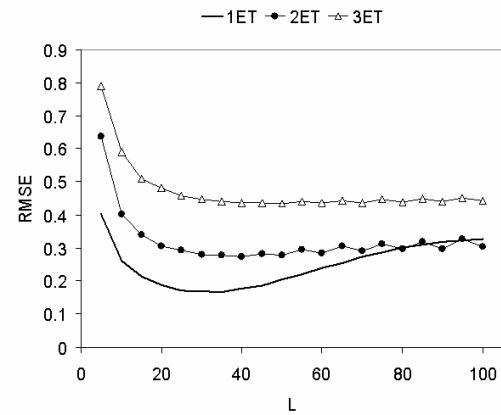


Рис. 3.31: RMSE восстановления:
в.р. (3.50)

3.4.8. SSA обработка стационарных временных рядов

Существуют специальные рекомендации по выбору параметров для стационарных временных рядов. Традиционно рекомендуется выполнять процедуру центрирования для стационарных временных рядов перед обработкой (т.е. вычесть среднее значение по временному

ряду) и затем использовать теплицеву автоковариационную матрицу $\tilde{\mathbf{C}}$ с элементами

$$\tilde{c}_{ij} = \frac{1}{N - |i - j|} \sum_{m=0}^{N-|i-j|-1} x_m x_{m+|i-j|}$$

вместо $\mathbf{C} = \mathbf{XX}^T$ на этапе разложения (подробности алгоритма Toeplitz SSA см. в разделе 1.8.6 и в [1]). Использование $\tilde{\mathbf{C}}$ не дает нам SVD траекторной матрицы и, следовательно, оптимальность SVD.

В работах, связанных с SSA-анализом климатических временных рядов (например, [168]), Toeplitz SSA рассматривают в качестве основной версии и утверждают, что базовая и теплицева версии отличаются лишь незначительно. Наше исследование показывает, что Toeplitz SSA обеспечивает более стабильные результаты, чем Basic SSA (восстановление, прогноз, оценки). Однако эти результаты могут быть неадекватными и иметь значительную погрешность, если анализируемый временной ряд не является стационарным. По-видимому, использование теплицевой версии алгоритма SSA небезопасно, если временной ряд содержит тренд или колебания с возрастающей или убывающей амплитудой.

Здесь мы можем применить хорошо известный принцип: если метод предполагает модель, то он дает более точные результаты, когда модель верна; в противном случае метод может дать совершенно неверные результаты.

Центрирование временного ряда — менее рискованная процедура, чем использование $\tilde{\mathbf{C}}$ и может либо немного улучшить результаты SSA, либо ухудшить их. Отметим, что центрирование обычно увеличивает ранг сигнала, делая его структуру более сложной.

Напомним, что кроме SSA с центрированием в качестве препроцессинга существует так называемый “Single centering SSA”, который подходит для временных рядов с постоянным трендом, а также так называемый “Double centering SSA”, который хорошо работает для временных рядов с линейными трендами (см. раздел 2.1.1).

В следующих подразделах мы демонстрируем примеры применения процедуры центрирования и алгоритма Toeplitz SSA к нестационарным временным рядам.

Центрирование как предварительная обработка

Пусть $N = 199$ и $x_n = s_n + r_n$ с

$$s_n = 1.005^n, \quad r_n = \varepsilon_n, \tag{3.51}$$

см. рис. 3.32.

Ранг экспоненциального ряда равен 1, поэтому для его извлечения мы должны выбрать только одну собственную тройку. После центрирования этого временного ряда мы получим новый $\mathbb{Y}_N = (y_1, \dots, y_N)$ с $y_n = 1.005^n - c + \varepsilon_n$. Поэтому для извлечения $1.005^n - c$ мы должны выбрать две собственных тройки, т.е. искусственно создать более сложную структуру сигна-

ла. Результат моделирования (см. рис. 3.33, L меняется от 5 до 100 с шагом 5; восстановление по ET1 выполняется без центрирования, а также восстановление по ET1–2 выполняется с предварительным центрированием) подтверждают, что сгущение структуры сигнала заканчивается ростом ошибок.

Практически, процедура центрирования может увеличить ошибки восстановления даже незатухающих синусоид в коротких временных рядах, если длина временного ряда не кратна периоду синусоиды. Объяснение аналогично объяснению для экспоненциального временного ряда: если длина временного ряда не кратна периоду синусоиды, то среднее значение по временному ряду не равно 0. Поэтому, вычитая это среднее значение мы преобразуем сигнал ранга 2 в сигнал ранга 3. Для длинных временных рядов эффект увеличения ранга уменьшается, поскольку среднее значение временного ряда почти равно нулю. Однако в этом случае центрирование не имеет смысла.

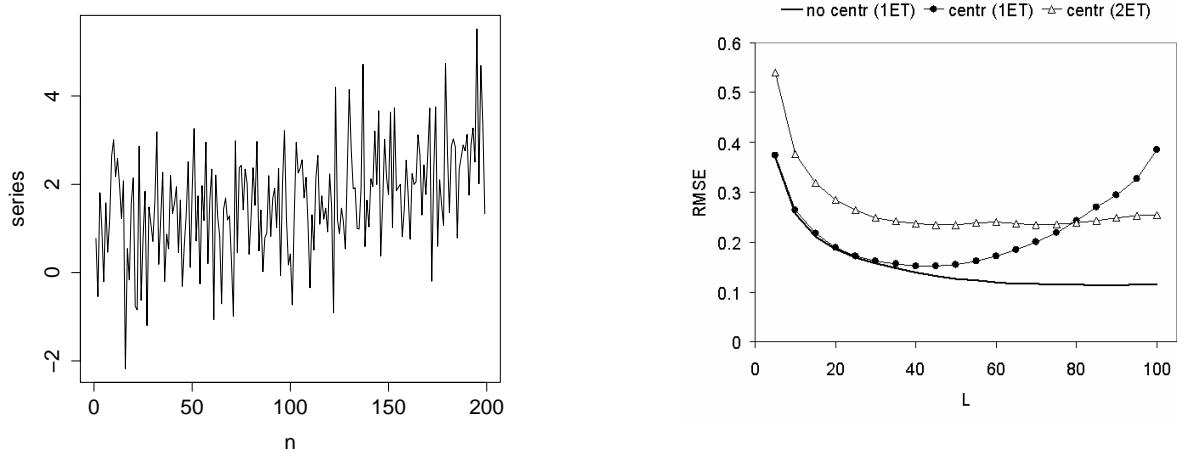


Рис. 3.32: Начальный временной ряд:
в.р. (3.51).

Рис. 3.33: RMSE восстановления:
в.р. (3.51).

Замечание 33 (О заполнении недостающих данных). Отметим, что метод заполнения недостающих значений, представленный в разделе 3.1, не предполагает центрирования (хотя центрирование может быть использовано). В методе из [133] процедура центрирования является необходимой, так как недостающие значения заменяются нулями на первой итерации. По-видимому, эта особенность вызвана тем, что первые применения SSA были ориентированы на стационарные временные ряды. Конечно, для обобщения метода, предложенного в [133], можно заполнить недостающие значения с помощью среднего значения по всему временному ряду или использовать метод интерполяции, основанный на соседних непропущенных значениях.

Замечание 34 (О собственных значениях). Собственные значения играют важную роль в анализе стационарных временных рядов (см., например, работы, посвященные Монте-Карло SSA для обнаружения сигнала в красном шуме [151]). Для стационарных временных рядов доля собственных троек, соответствующих сигналу, представляет собой вклад соответствующего компонента. Однако в общем случае произвольных временных рядов доля собственных троек имеет мало смысла, поскольку она зависит от абсолютных значений временного ряда: в Basic SSA распределение собственных значений (в основном, ведущих) зависит от постоянного слагаемого временного ряда, в то время как структура временного ряда не должна зависеть от сложения или вычитания константы. В частности, если ведущая собственная величина занимает 99,9%, это не означает, что достаточно взять только одну собственную тройку, чтобы аппроксимировать временной ряд с высокой точностью. Конечно, мы можем использовать информацию о собственных значениях, чтобы определить пары собственных троек, порожденных синусоидами (одна синусоида порождает два близких или одинаковых собственных значения). Однако такая же идентификация может быть выполнена с помощью более мощных инструментов.

Toeplitz SSA

Давайте продемонстрируем последствия неправильного использования Toeplitz SSA. Во-первых, применение Toeplitz SSA к нестационарным сигналам конечного ранга r обычно увеличивает число ненулевых собственных значений с r до максимально возможного значения равного $\min(L, K)$, то есть структура может быть потеряна. Это означает, что количество собственных значений необходимых для точного восстановления, увеличивается. Кроме того, построенная аппроксимация может иметь неправильную структуру и, например, привести к неверному прогнозу.

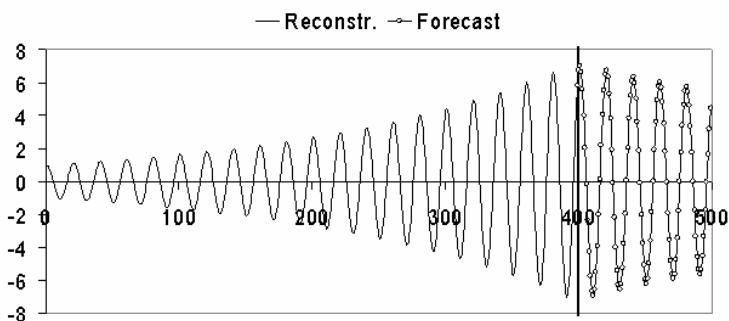


Рис. 3.34: Прогноз Toeplitz SSA для $x_n = 1.005^n \cos(2\pi n/20)$, $N = 399$, $L = 200$, ET1–14

Мы рассмотрим два примера временных рядов конечного ранга, $x_n = 1.005^n \cos(2\pi n/20)$ (ранг 2) и $x_n = 1.005^n$ (ранг 1). Для получения точного приближения последних точек временного ряда, мы выполняем восстановление Toeplitz-SSA с $L = 200$ на основе ET1–14 (ET1–2 для первого примера и ET1 для второго не обеспечивают хорошей точности). Рис. 3.34 и 3.35 показывают, что прогноз, основанный на выбранных компонентах, неадекватен (более того,

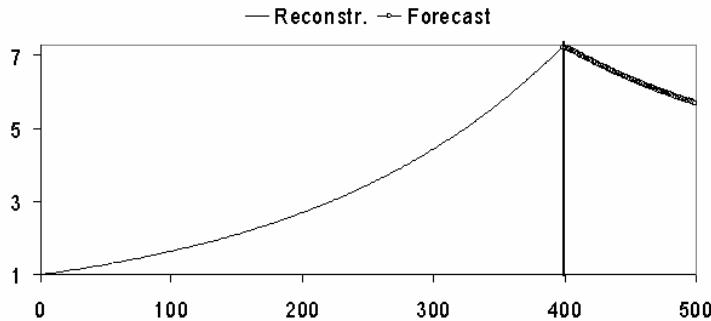


Рис. 3.35: Прогноз Toeplitz SSA для $x_n = 1.005^n$, $N = 399$, $L = 200$, ET1–14

этот вывод не зависит от выбранной длины окна L и количества компонент).

Видно, что прогноз Toeplitz-SSA полностью неправильный, в то время как прогноз Basic SSA для временных рядов конечного ранга является точным.

3.4.9. SVD-источники SSA и выбор параметров SSA

Ключевым шагом в SSA является сингулярное разложение (SVD) траекторной матрицы. SVD используется для решения различных задач, включая статистические методы в анализе данных. Поэтому логика этих процедур может быть распространена и на SSA. Кратко опишем несколько истоков идей, связанных с SVD, и порожденные ими взгляды на выбор SSA-параметров. Мы не ставим целью обзор литературы по SVD, а лишь подчеркиваем связь между истоками и методологией SSA.

Анализ главных компонент (PCA) Это происхождение характеризуется различными действиями с признаками и наблюдениями в данных, т.е. со строками и столбцами траекторной матрицы. В частности, традиционными действиями являются центрирование и стандартизация переменных. Это приводит к различным взглядам на собственные векторы и факторные векторы (векторы главных компонент), где последние интерпретируются как компоненты исходного временного ряда (возможно, чуть более короткие, чем исходный временной ряд). Кроме того, из PCA в SSA переносится большее внимание к вкладу собственных значений. Такое отношение к траекторной матрице уместно, когда мы применяем SSA с однократным центрированием и когда количество строк (L) фиксировано и меньше количества столбцов (K). В общем случае структура траекторной матрицы не зависит от транспонирования траекторной матрицы, интерпретация собственных значений не так важна и, в частности, двойное центрирование часто более естественно, чем однократное.

Ганкелевы матрицы неполного ранга Связь между такими матрицами и временными рядами, управляемыми линейными рекуррентными соотношениями, известна давно (см., например, [80]). Эта техника позволяет нам анализировать зашумленные временные ряды, управляемые ЛРС. Основное применение этой идеи относится к области обработки сигналов с ее подходами к выбору параметров метода и заключается в анализе зашумленной суммы затухающих/незатухающих цисоид и оценке их параметров (в основном, частот). Однако этот

подход может быть применен для оценки параметров произвольных сигналов, управляемых ЛРС.

Спектральный анализ [79] Название этого подхода близко к Сингулярному спектральному анализу. Однако название ‘Спектральный анализ’ означает анализ стационарных временных рядов и их частотных характеристик. С другой стороны, сингулярный спектр связан со спектром линейных операторов, т.е. сингулярные значения траекторных матриц в случае SSA. Таким образом, анализ сингулярного спектра не подразумевает стационарности. Как было указано в [65], название SSA не отражает многогранной сущности SSA и используется традиционно. Особенность этого происхождения заключается в центрировании временного ряда перед обработкой и последующим применением теплицевой версии SSA. Как было показано в разделе 3.4.8, если применить эту технику к нестационарным временным рядам, то, скорее всего, мы получим неточные или даже бессмысличные результаты. В рамках подхода спектрального анализа особое внимание уделяется красному шуму (авторегрессии порядка 1) и проверке нулевой гипотезы об отсутствии сигнала в красном шуме. В некотором смысле это соответствует задаче со слабым сигналом и сильным шумом.

Разложение Карунена-Лоэва [76] Это разложение традиционно используется в теории стохастических процессов и исходно предполагает нулевое ожидание элементов рассматриваемого стохастического процесса (или вычитание известных средних). В [76] оценка среднего значения процесса производится с помощью скользящего среднего. Затем остатки от этой оценки включаются в центрированный стохастический процесс, что позволяет применять метод к процессам с трендами. Таким образом, алгоритм оказывается очень близким к PCA и фактически совпадает с SSA with single centering. Похоже, что это происхождение SSA используется исследователями, которые хорошо знакомы с методами стохастических процессов, и поэтому ассоциация с KL-разложением помогает им понять метод SSA.

Динамические системы [60, 61] Это происхождение связано со специальными проблемами в теории динамических систем со специфическим подходом к выбору параметров. Однако вклад этих работ значителен, поскольку описанный алгоритм послужил зарождением идей SSA в нескольких прикладных областях.

Глава 4

SSA для многомерных временных рядов

В этой главе мы рассматриваем задачу одновременного разложения, восстановления и прогнозирования для набора временных рядов посредством SSA. Основной метод этой главы обычно называется Multichannel SSA или Multivariate SSA, сокращенно MSSA. Основная идея алгоритма та же, что и для Basic SSA, разница заключается в том, как строится траекторная матрица. Цель MSSA состоит в том, чтобы принять во внимание общую структуру многомерного ряда для получения более точных результатов.

MSSA обычно рассматривается как расширение 1D-SSA. Однако алгоритм MSSA был опубликован даже раньше, чем алгоритм 1D-SSA; см. [169], где MSSA назывался анализом расширенных эмпирических ортогональных функций (Extended empirical orthogonal functions, EEOF). Алгоритм MSSA в рамках SSA был формально сформулирован в [170]. Здесь мы рассматриваем алгоритм MSSA для анализа и прогнозирования многомерных временных рядов, следуя подходу, описанному в [1, Глава 2] для одномерных рядов и в [35] для многомерных.

В то время как существует более или менее каноническое описание 1D-SSA, алгоритмы анализа и прогнозирования по MSSA значительно отличаются (хотя и эквивалентны). Поэтому в разделах 4.1 и 4.3 мы рассматриваем различные обозначения и выписываем версии, которые мы считаем каноническими согласно [12] и [31].

В разделе 4.1 мы расширяем методологию главы 2 для SSA-анализа системы нескольких временных рядов. Важно отметить, что существует два основных способа упаковки отдельных траекторных матриц в общую траекторную матрицу: горизонтальная укладка и вертикальная укладка. Для анализа MSSA эти две процедуры упаковки эквивалентны.

В разделе 4.2 мы собрали части теории, связанные с разделимостью, рангами и формой разложения синусоид, которая используется в автоматической группировке.

В разделе 4.3 рассматривается прогнозирование в MSSA. Существует четыре основных варианта MSSA прогнозирования: рекуррентное прогнозирование по столбцам, рекуррентное прогнозирование по строкам, векторное прогнозирование по столбцам и векторное прогнозирование по строкам. Мы тщательно описываем общие черты и различия между этими четырьмя вариантами и проводим их сравнение на модельных данных.

Раздел 4.4 содержит расширения методов автоматической группировки, которые описаны в разделе 2.5 для одномерных временных рядов.

4.1. MSSA анализ

4.1.1. Метод

Рассмотрим многомерный временной ряд; то есть, набор $\{\mathbf{X}^{(p)} = (x_j^{(p)})_{j=1}^{N_p}, p = 1, \dots, s\}$ из s временных рядов длины N_p , $p = 1, \dots, s$.

Обозначим $\mathbf{X} = (\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(s)})$ начальные данные для MSSA алгоритма. Общая схема алгоритма, описанная в разделе 1.2, применима и к MSSA; нам нужно только определить оператор вложения $\mathcal{T}_{\text{MSSA}}(\mathbf{X}) = \mathbf{X}$.

Вложение Пусть L — целое число, называемое длиной окна, $1 < L < \min(N_p, p = 1, \dots, s)$. Для каждого временного ряда $\mathbf{X}^{(p)}$ мы формируем $K_p = N_p - L + 1$ L -запаздывающих векторов $X_j^{(p)} = (x_j^{(p)}, \dots, x_{j+L-1}^{(p)})^T$, $1 \leq j \leq K_p$. Обозначим $K = \sum_{p=1}^s K_p$. Траекторная матрица из многомерного ряда \mathbf{X} является матрицей размера $L \times K$ и имеет вид

$$\mathcal{T}_{\text{MSSA}}(\mathbf{X}) = \mathbf{X} = [X_1^{(1)} : \dots : X_{K_1}^{(1)} : \dots : X_1^{(s)} : \dots : X_{K_s}^{(s)}] = [\mathbf{X}^{(1)} : \dots : \mathbf{X}^{(s)}], \quad (4.1)$$

где $\mathbf{X}^{(p)} = \mathcal{T}_{\text{1D-SSA}}(\mathbf{X}^{(p)})$ — траекторная матрица одномерного ряда $\mathbf{X}^{(p)}$, определенная в (1.1). Таким образом, траекторная матрица системы временных рядов состоит из *состыкованных ганкелевых матриц*. Заметим, что

$$\mathcal{T}_{\text{MSSA}}^{-1}(\mathbf{X}) = [\mathcal{T}_{\text{SSA}}^{-1}(\mathbf{X}^{(1)}) : \dots : \mathcal{T}_{\text{SSA}}^{-1}(\mathbf{X}^{(s)})]. \quad (4.2)$$

Комментарии к форме шага вложения В MSSA (напомним, что MSSA — это аббревиатура от Multivariate/Multichannel SSA) траекторная матрица строится из состыкованных траекторных матриц временных рядов из рассматриваемого набора. Укладка этих матриц может быть как вертикальной, так и горизонтальной. Мы рассматриваем горизонтальную укладку как основной вариант. В некоторых работах, включая работы по климатологическим приложениям MSSA (см., например, [60, 152, 171]), траекторная матрица транспонируется, т.е., траекторные матрицы $\mathbf{X}^{(p)}$ укладываются вертикально.

В случае вертикальной укладки (при равных длинах рядов),

$$\mathcal{T}_{\text{MSSA}}(\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(s)}) = \begin{pmatrix} \mathbf{X}^{(1)} \\ \vdots \\ \mathbf{X}^{(s)} \end{pmatrix}. \quad (4.3)$$

Вертикальная укладка принципиально не отличается от горизонтальной; левый и правый сингулярные векторы меняются местами (после замены друг на друга L и $N - L + 1$) и поэтому способ укладывания влияет только на терминологию, выбор параметров и, возможно, на вычислительные затраты. Важно иметь в виду, что в литературе по MSSA существует множество различной терминологии, и некоторые термины из них несколько противоречивы.

При горизонтальной укладке мы фиксируем количество строк, то есть размерность пространства столбцов траекторной матрицы. В этом случае увеличение длины временного ряда приводит к увеличению размерности столбцов траекторной матрицы. Напомним, что при горизонтальной укладке длины разных временных рядов могут отличаться, и это не влияет на размерность столбцов.

Для простой иллюстрации рассмотрим горизонтальную и вертикальную укладку для двух временных рядов, $X^{(1)} = (11, 12, 13, 14, 15)$ и $X^{(2)} = (21, 22, 23, 24)$. Для горизонтальной укладки с $L = 2$ мы имеем

$$\left(\begin{array}{cccc|ccc} 11 & 12 & 13 & 14 & 21 & 22 & 23 \\ 12 & 13 & 14 & 15 & 22 & 23 & 24 \end{array} \right).$$

Вертикальная укладка возможно только при одинаковом количестве столбцов в отдельных траекторных матрицах; поэтому в нашем примере мы должны использовать разные длины окна для разных временных рядов; зададим $L_1 = 2$ и $L_2 = 1$:

$$\left(\begin{array}{cccc} 11 & 12 & 13 & 14 \\ \hline 12 & 13 & 14 & 15 \\ \hline 21 & 22 & 23 & 24 \end{array} \right).$$

Объясним разницу между вертикальной постановкой в форме (4.3) и в форме, используемой в [169]. Пусть $X^{(j)} = (j1, j2, j3, j4, j5)$, где $j = 1, 2, 3$ — номера некоторых географических точек, вторая цифра соответствует времени. Тогда строки траекторной матрицы \mathbf{X} для MSSA с вертикальной укладкой упорядочены по номерам рядов, тогда как строки траекторной матрицы \mathbf{X}_{WN} , использованной в [169], упорядочены по времени:

$$\mathbf{X} = \left(\begin{array}{cccc} 11 & 12 & 13 & 14 \\ 12 & 13 & 14 & 15 \\ \hline 21 & 22 & 23 & 24 \\ 22 & 23 & 24 & 25 \\ \hline 31 & 32 & 33 & 34 \\ 32 & 33 & 34 & 35 \end{array} \right), \quad \mathbf{X}_{WN} = \left(\begin{array}{cccc} 11 & 12 & 13 & 14 \\ 21 & 22 & 23 & 24 \\ \hline 31 & 32 & 33 & 34 \\ 12 & 13 & 14 & 15 \\ \hline 22 & 23 & 24 & 25 \\ 32 & 33 & 34 & 35 \end{array} \right). \quad (4.4)$$

Однако SVD-разложения двух траекторных матриц в (4.4) фактически одинаковы и отличаются только упорядоченностью компонент вектора. Левые сингулярные векторы размером $6 = 2 \times 3$ можно разделить на две части, сдвинутые во времени. Каждая часть состоит из трех географических точек. В [169] левые сингулярные векторы длины L_s называются EEOFs (расширенные эмпирические ортогональные функции). Каждая EEOF делится на две части, соответствующие двум временным лагам (число лагов равно $L = 2$), а затем каждая часть изображается в виде поверхности.

Разложение Обычное разложение на матрицы ранга один на этапе разложения в MSSA строится путем применения SVD к траекторной матрице; то есть, стандартный вариант MSSA является расширением Basic SSA и поэтому его можно назвать Basic MSSA.

Косоугольные модификации MSSA такие же, как и в одномерном случае; то есть, можно строить вложенные разложения с помощью Iterative O-SSA и Filter-adjusted O-SSA. Использование этих вложенных модификаций точно такое же, как и в одномерном случае и мы отсылаем читателя к разделам 2.2 и 2.3.1 за подробностями.

Восстановление Поскольку $\mathcal{M}_{L,K}^{(H)}$ в MSSA является множеством состыкованных ганкелевых матриц, ортогональный проектор $\Pi_{\text{stacked }} \mathcal{H}$ на $\mathcal{M}_{L,K}^{(H)}$ имеет вид

$$\Pi_{\text{stacked }} \mathcal{H}(\mathbf{Y}) = [\Pi_{\mathcal{H}}(\mathbf{Y}^{(1)}) : \dots : \Pi_{\mathcal{H}}(\mathbf{Y}^{(s)})], \quad (4.5)$$

где $\Pi_{\mathcal{H}}$ — проектор на множество ганкелевых матриц. Равенство (4.5) следует из общей формы проекции, описанной в разделе 1.2.2. Как и в одномерном случае, восстановленные ряды получаются путем композиции $\mathcal{T}_{\text{MSSA}}^{-1}$ и $\Pi_{\text{stacked }} \mathcal{H}$.

Комментарии Сделаем несколько замечаний относительно характерных особенностей метода MSSA.

Собственные векторы $\{U_i\}$ в SVD траекторной матрицы $\mathbf{X} = \sum_i \sqrt{\lambda_i} U_i V_i^T$ образуют общий базис траекторных пространств столбцов всех временных рядов системы. Факторные векторы $\{V_i\}$ состоят из частей, относящихся к каждомуциальному ряду отдельно; а именно,

$$V_i = \begin{pmatrix} V_i^{(1)} \\ \vdots \\ V_i^{(s)} \end{pmatrix}, \quad (4.6)$$

где p -ый факторный подвектор $V_i^{(p)} \in \mathbb{R}^{K_p}$ принадлежит траекторному пространству p -го ряда.

Собственные векторы U_i отражают общие черты временных рядов, в то время как факторные подвекторы $V_i^{(p)}$ показывают, как эти общие черты проявляются в каждом ряде. Естественно представлять факторные векторы в виде систем факторных подвекторов $V_i^{(p)}$. Тогда форма преобразованных факторных векторов будет аналогична исходной системе рядов.

Аналогично одномерному случаю основным результатом применения MSSA является разложение многомерного временного ряда в сумму m многомерных рядов; параметрами являются длина окна L и способ группировки. Для часто используемого случая двух групп мы обозначим через $\tilde{\mathbf{X}}^{(k)} = (\tilde{x}_j^{(k)})_{j=1}^N$, $k = 1, \dots, s$, восстановленный ряд (обычно, сигнал), соответствующий первой группе собственных троек I_1 .

Замечание 35. 1. Индексация временных точек $1, \dots, N_p$ ($p = 1, \dots, s$), начиная с 1, не означает, что все ряды s начинаются в одинаковое время; они также могут заканчиваться в разное время. Результирующее разложение, полученное алгоритмом MSSA, не зависит от сдвига между одномерными рядами, и поэтому эта индексация является лишь формальностью. В частности, MSSA-разложение двух одномерных рядов, измеренных в один и тот же временной интервал и в разные временные интервалы, не различаются.

2. Исходные временные интервалы для рядов $X^{(p)}$ могут быть полезны для их изображения и интерпретации. Восстановленные ряды имеют те же временные диапазоны, что и исходные ряды. Факторные субвекторы из факторной системы также могут быть синхронизированы для построения графика на основе диапазонов исходных рядов; хотя факторные векторы короче исходных рядов, их временные сдвиги одинаковы.
3. Для одновременного анализа нескольких временных рядов рекомендуется переводить их в один и тот же масштаб. В противном случае структура одного конкретного временного ряда будет оказывать слишком большое влияние на результаты MSSA. Чтобы уравновесить временные ряды, их можно либо стандартизировать (центрировать и нормализовать, в аддитивных моделях), либо только нормализовать (в мультипликативных моделях). С другой точки зрения, масштабирование отдельных рядов может быть использовано для влияния на важность конкретного ряда системы, когда, например, какой-то конкретный ряд более важен или имеет меньшую шумовую составляющую.
4. Алгоритм MSSA может быть модифицирован теми же способами, что и алгоритм 1D-SSA. Например, можно рассмотреть Toeplitz MSSA и MSSA с проекцией (включая центрирование). (Однако, эти опции не реализованы в текущей версии R-пакета RSSA). Вложенные косоугольные вариации 1D-SSA (Iterative O-SSA и Filter-adjusted O-SSA) реализованы в RSSA.
5. В климатологии традиционно рассматривается [171] SVD транспонированной (вертикально сложенной) траекторной матрицы, определенной в (4.1). Поэтому собственные векторы $\{U_i\}$ соответствуют нормализованным расширенным главным компонентам в [171], а факторные векторы $\{V_i\}$ называются EOF в климатологических приложениях, начиная с [169].
6. Вычислительная стоимость SVD на шаге разложения зависит от размера матрицы \mathbf{XX}^T и, следовательно, эти вычислительные затраты могут существенно отличаться для версий MSSA с горизонтальной и вертикальной укладкой траекторных матриц.

4.1.2. Алгоритм

Алгоритм разложения в MSSA отличается от алгоритма 1.1 разложения в Basic SSA только формой оператора вложения.

АЛГОРИТМ 4.1: MSSA: разложение

Input: Набор $\{\mathbf{X}^{(p)} = (x_j^{(p)})_{j=1}^{N_p}, p = 1, \dots, s\}$ из s временных рядов длины N_p , $p = 1, \dots, s$, длина окна L .

Output: Разложение траекторной матрицы по элементарным матрицам $\mathbf{X} = \mathbf{X}_1 + \dots + \mathbf{X}_d$, где $\mathbf{X}_i = \sqrt{\lambda_i} U_i V_i^T$.

- 1: Построить траекторную матрицу $\mathbf{X} = \mathcal{T}_{\text{MSSA}}(\mathbf{X})$, где $\mathcal{T}_{\text{MSSA}}$ определяется в (4.1).
- 2: Вычислить SVD $\mathbf{X} = \mathbf{X}_1 + \dots + \mathbf{X}_d$, $\mathbf{X}_i = \sqrt{\lambda_i} U_i V_i^T$.

Этап восстановления также очень похож на алгоритм 1.2 для восстановления в Basic SSA.

АЛГОРИТМ 4.2: MSSA восстановление

Input: Разложение $\mathbf{X} = \mathbf{X}_1 + \dots + \mathbf{X}_d$, где $\mathbf{X}_i = \sigma_i U_i V_i^T$ и $\|U_i\| = \|V_i\| = 1$, группировка $\{1, \dots, d\} = \bigsqcup_{j=1}^m I_j$.

Output: Разложение системы временных рядов на идентифицируемые компоненты $\mathbf{X} = \mathbf{X}_1 + \dots + \mathbf{X}_m$.

- 1: Построить сгруппированное разложение матрицы $\mathbf{X} = \mathbf{X}_{I_1} + \dots + \mathbf{X}_{I_m}$, где $\mathbf{X}_I = \sum_{i \in I} \mathbf{X}_i$.
- 2: $\mathbf{X} = \mathbf{X}_1 + \dots + \mathbf{X}_m$, где $\mathbf{X}_i = \mathcal{T}_{\text{MSSA}}^{-1} \circ \Pi_{\text{stacked}} \mathcal{H}(\mathbf{X}_{I_i})$.

Напомним, что $\mathcal{T}_{\text{MSSA}}$ и $\Pi_{\text{stacked}} \mathcal{H}$ могут быть выражены через \mathcal{T}_{SSA} и $\Pi_{\mathcal{H}}$, введенные в главе 2 для 1D-SSA (см. раздел 4.1.1).

4.2. Элементы теории MSSA

Мы помещаем в этот раздел несколько предложений, связанных с теорией MSSA [33, 34].

4.2.1. Разделимость

Разделимость является ключевым понятием в теории SSA, так как разделимость рядов означает способность метода извлекать их из заданного суммарного ряда. Понятие разделимости для многомерных временных рядов аналогично понятию для одномерных рядов, которое кратко прокомментировано в разделе 1.2.2 и подробно описано в [1, Разделы 1.5 и

6.1]. В нем также существует слабая разделимость, которая означает ортогональность траекторных пространств, и сильная разделимость, которая означает пустое пересечение множеств сингулярных значений, порожденных разделяемым рядами.

Как правило, условия разделимости многомерных временных рядов более ограничительные, чем для одномерных рядов. Ниже приведено достаточное условие слабой разделимости.

Предложение 19. *Если временные ряды $F^{(1)}$ и $F^{(2)}$, $G^{(1)}$ и $G^{(2)}$, $F^{(1)}$ и $G^{(2)}$, а также $G^{(1)}$ и $F^{(2)}$ слабо L -разделимы по SSA, тогда двумерные временные ряды $(F^{(1)}, F^{(2)})$ и $(G^{(1)}, G^{(2)})$ слабо L -разделимы по MSSA.*

Доказательство непосредственно следует из определения разделимости.

Предложение 19 может быть распространено на аналогичный результат для асимптотической разделимости ($N_i \rightarrow \infty$) и, следовательно, для приближенной разделимости при фиксированных N_i .

Следующий пример является следствием предложения 19 с учетом известных условий разделимости одномерных рядов.

Пример 2. Рассмотрим пример четырех гармонических вещественных временных рядов $F^{(1)}$, $F^{(2)}$, $G^{(1)}$ и $G^{(2)}$ длины N :

$$f_k^{(1)} = A_1 \cos(2\pi\omega_1 k + \varphi_1), \quad f_k^{(2)} = B_1 \cos(2\pi\omega_1 k + \varphi_2),$$

$$g_k^{(1)} = A_2 \cos(2\pi\omega_2 k + \phi_1), \quad g_k^{(2)} = B_2 \cos(2\pi\omega_2 k + \phi_2),$$

$\omega_1 \neq \omega_2$, $k = 0, \dots, N - 1$, $A_1, A_2, B_1, B_2 \neq 0$. Если $L\omega_i$ и $K\omega_i$, $i = 1, 2$, являются целыми, то $(F^{(1)}, F^{(2)})$ и $(G^{(1)}, G^{(2)})$ слабо L -разделимы по MSSA.

Заметим, что если $L\omega_i$ или $K\omega_i$ не целое число, то ряды становятся приближенно разделимыми.

Слабой разделимости недостаточно для выделения компонентов временного ряда. Поэтому давайте рассмотрим сильную разделимость, связанную с собственными значениями создаваемых компонентами временного ряда. Оказывается, что каждый временной ряд, многомерный или отдельный, может давать разные собственные значения в SSA и MSSA. Поэтому применение многомерной модификации SSA может улучшить (или ухудшить) сильную разделимость.

4.2.2. Ранги и подпространства

Рассмотрим систему из s бесконечных временных рядов $X = (X^{(1)}, X^{(2)}, \dots, X^{(s)})$, выберем длину окна L и обозначим за $\mathcal{X}^{(1)}, \dots, \mathcal{X}^{(s)}$ столбцовые траекторные пространства столбцов рядов. Пусть $\mathcal{X} = \text{span}(\mathcal{X}^{(1)}, \dots, \mathcal{X}^{(s)})$ является столбцовным траекторным пространством набора

временных рядов X . Как и для одномерных временных рядов, мы называем размерность траекторного пространства (равную рангу траекторной матрицы набора рядов) рангом набора рядов, см. раздел 1.2.2 для краткого описание общих понятий.

Обозначим ранги $X^{(l)}$ через $r_l = \dim X^{(l)} \leq L$, $l = 1, \dots, s$. Для каждого временного ряда $X^{(l)}$ мы можем выписать минимальное ЛРС, управляющее этим временным рядом:

$$x_{j+r_l}^{(l)} = \sum_{k=1}^{r_l} a_k^{(l)} x_{j+r_l-k}^{(l)}, \quad \text{где } a_{r_l}^{(l)} \neq 0, \quad l = 1, \dots, s. \quad (4.7)$$

Соответствующие характеристические многочлены ЛРС (4.7) имеют вид

$$P_{r_l}^{(l)}(\mu) = \mu^{r_l} - \sum_{k=1}^{r_l} a_k^{(l)} \mu^{r_l-k}, \quad l = 1, \dots, s. \quad (4.8)$$

Напомним, что корни характеристического многочлена минимального ЛРС, управляющего рядом, называются характеристическими корнями.

Пусть

$$\begin{aligned} p^{(l)} & \text{число различных корней многочлена } P_{r_l}^{(l)}(\lambda), \\ \mu_m^{(l)} & m\text{-ый корень многочлена } P_{r_l}^{(l)}(\lambda), \\ k_m^{(l)} & \text{кратность корня } \mu_m^{(l)}. \end{aligned}$$

Тогда, из стандартной теории временных рядов конечного ранга [1], мы имеем, что

$$k_1^{(l)} + \dots + k_p^{(l)} = r_l, \quad l = 1, \dots, s.$$

Характеристические корни определяют поведение ряда. Например, если $k_m^{(l)} = 1$, то временной ряд имеет вид

$$x_n^{(l)} = \sum_{j=1}^{r_l} C_j^{(l)} (\mu_j^{(l)})^n.$$

Также пусть

$$\begin{aligned} \mu_1, \dots, \mu_p & \text{объединенное множество корней всех многочленов } P_{r_1}^{(1)}, \dots, P_{r_s}^{(s)}, \\ k_1, \dots, k_p & \text{множество корней } \mu_1, \dots, \mu_p, \end{aligned}$$

где кратность корня в объединенном множестве равна максимальной кратности соответствующего корня во всех многочленах.

Поскольку корни определяются структурой траекторного пространства, можно доказать следующее утверждение.

Предложение 20. В введенных выше обозначениях MSSA-ранг многомерного временного ряда $X = (X^{(1)}, X^{(2)}, \dots, X^{(s)})$ равен $r = \sum_{i=1}^p k_i$, для $L > r$.

Доказательство. Обозначим для комплексной величины $\mu = \rho e^{i2\pi\omega}$, $-1/2 < \omega \leq 1/2$, и $j \geq 0$

$$Q(\mu, j) = \begin{pmatrix} \mu^0 0^j \\ \mu^1 1^j \\ \vdots \\ \mu^{L-1} (L-1)^j \end{pmatrix}, \quad P(\mu, j) = \begin{cases} Q(\mu, j), & \text{если } \omega = 0 \text{ или } \omega = 1/2, \\ \Re(Q(\mu, j)), & \text{если } 0 < \omega < 1/2, \\ \Im(Q(\mu, j)), & \text{если } -1/2 < \omega < 0. \end{cases}$$

Здесь $\Re(X)$ и $\Im(X)$ — действительная и мнимая части X , $Q(\mu, j) \in \mathbb{C}^L$ и $P(\mu, j) \in \mathbb{R}^L$. По построению, векторы $P(\mu_m, j)$, где $m = 1, \dots, p$, $j = 0, \dots, k_m^{(l)} - 1$, являются образующими траекторного пространства \mathcal{X} матрицы \mathbf{X} . Поскольку эти вектора линейно независимы при $L > r$, доказательство закончено. \square

Согласованность рядов Одновременный анализ нескольких временных рядов обычно проводится для выявления их взаимосвязи и выявления их общей структуры. Напомним, что для 1D-SSA временной ряд имеет структуру тогда и только тогда, когда траекторная матрица этого ряда имеет неполный ранг. Конечно, для типичного реального ряда, траекторная матрица имеет полный ранг. Поэтому в дальнейшем мы будем говорить о ранге сигнала (составляющей временного ряда со структурой) или его компонентов.

Рассмотрим систему сигналов $\mathbf{H} = (\mathbf{H}^{(1)}, \mathbf{H}^{(2)})$ с траекторной матрицей неполного ранга. Структура ряда отражается в его траекторном пространстве. Мы можем сказать, что два временных ряда имеют одинаковую структуру, если их траекторные пространства совпадают. Например, для двух синусоид с одинаковыми периодами их траекторные пространства совпадают, независимо от значений их амплитуд и фаз. Это следует из того, что траекторное пространство — это совокупность подрядов длины L исходного ряда. С другой стороны, синусоиды с разными частотами имеют совершенно разную структуру, и объединенное траекторное пространство их системы является прямой суммой траекторных пространств рядов.

Если два временных ряда полностью совпадают, то траекторное пространство одного временного ряда может быть использовано для восстановления или прогнозирования второго ряда. Если два ряда не связаны между собой и имеют совершенно разную структуру, то ни один из них не содержит никакой полезной информации о другом ряде для анализа MSSA.

Для MSSA любой сдвиг между двумя временными рядами и любая разница между фазами двух согласованных синусоид не влияет на результат анализа. Поэтому нельзя сказать что-либо о направлении причинно-следственной связи. Более того, асимметрия влияния одного временного ряда на другой может быть вызвана различными уровнями шума. Однако временной ряд $\mathbf{X}^{(2)}$ можно назвать *поддерживающим* для временного ряда $\mathbf{X}^{(1)} = \mathbf{H}^{(1)} + \mathbf{R}^{(1)}$, если точность восстановления или прогнозирования $\mathbf{H}^{(1)}$ улучшается, если мы анализируем систему двух рядов $\mathbf{X} = (\mathbf{X}^{(1)}, \mathbf{X}^{(2)})$ а не только ряд $\mathbf{X}^{(1)}$.

Численные эксперименты подтверждают, что для двух совпадающих сигналов ряд с

любым уровнем шума, который не больше другого, всегда поддерживающий (см., например, раздел 4.3.3).

4.2.3. Разложение э.-м. гармоник

Рассмотрим многомерный экспоненциально модулированный (э.-м.) гармонический ряд $S = (S^{(1)}, \dots, S^{(s)})$, где элементы p -го ряда имеют вид

$$s^{(p)}(n) = e^{\alpha n} a_p \cos(2\pi\omega n + \phi_p), \quad (4.9)$$

$0 \leq \phi_p < 2\pi$, $a_p \neq 0$, $0 < \omega \leq 0.5$, $\sin \phi_p \neq 0$ для $\omega = 0.5$, $n = 1, \dots, N_p$, $p = 1, \dots, s$.

Обозначим L длину окна, $K_p = N_p - L + 1$, $K = \sum_{p=1}^s K_p$, $\Lambda^{(1)}$ и $\Lambda^{(2)}$ — подпространства строк и столбцов траекторной матрицы многомерного ряда S .

Предложение 21. 1. L -ранг d ряда S , элементы которого заданы (4.9), равен 1, если $\omega = 0.5$; в остальных случаях $d = 2$.

2. Если $\omega \neq 0.5$, то подпространство $\Lambda^{(1)}$ имеет базис

$$\{(1, e \cos(2\pi\omega), \dots, e^{\alpha(L-1)} \cos(2\pi(L-1)\omega))^T, (0, e \sin(2\pi\omega), \dots, e^{\alpha(L-1)} \sin(2\pi(L-1)\omega))^T\}.$$

3. Если $\omega \neq 0.5$, то подпространство $\Lambda^{(2)}$ имеет базис

$$\{(c_1^{(1)}, \dots, c_{K_1}^{(1)}; \dots; c_1^{(s)}, \dots, c_{K_s}^{(s)})^T, (d_1^{(1)}, \dots, d_{K_1}^{(1)}; \dots; d_1^{(s)}, \dots, d_{K_s}^{(s)})^T\},$$

$\varepsilon \partial e$

$$c_j^{(p)} = a_p \cos(2\pi j\omega + \phi_p) \quad u \quad d_j^{(p)} = a_p \sin(2\pi j\omega + \phi_p), \\ j = 1, \dots, K_p, \quad p = 1, \dots, s.$$

4. В случае $d = 1$, одномерные подпространства $\Lambda^{(1)}$ и $\Lambda^{(2)}$ порождаются векторами $(1, c, \dots, c^{L-1})^T$ и $(1, c, \dots, c^{K-1})^T$ соответственно, где $c = -e^\alpha$.

Предложение 22. Пусть $\alpha = 0$, $\omega \neq 0.5$, $L\omega$ и $K\omega$ — целые числа, $s = 2$. Тогда собственные значения SVD траекторной матрицы двумерного ряда S , элементы которого заданы (4.9), одинаковы и имеют вид $\lambda_1 = \lambda_2 = (a^2 + b^2)LK/4$.

Пример 3. Пусть

$$s_k^{(1)} = A \cos(2\pi\omega k + \varphi_1), \quad s_k^{(2)} = B \cos(2\pi k\omega k + \varphi_2).$$

Если $L\omega$ и $K\omega$ целые, то $(S^{(1)}, S^{(2)})$ дает два одинаковых собственных значения в MSSA: $\lambda_1 = \lambda_2 = (A^2 + B^2)LK/4$. Заметим, что ряд $S^{(1)}$ сам по себе производит два собственных значения равных $A^2LK/4$.

Доказательство. 1. Обозначим $\psi = \varphi_1/2$. Тогда $\mathbf{X} = P_1 Q_1^\top + P_2 Q_2^\top$, где

$$\begin{aligned} p_{1k} &= \cos(2\pi\omega(k-1) + \psi), & p_{2k} &= -\sin(2\pi\omega(k-1) + \psi), \\ q_{1m} &= A \cos(2\pi\omega(m-1) + \psi), & q_{2m} &= A \sin(2\pi\omega(m-1) + \psi), \end{aligned}$$

для $1 \leq k \leq L$, $1 \leq m \leq K$.

Пусть $p = \|P_1\|/\|P_2\|$, $q = \|Q_1\|/\|Q_2\|$, $S = \|P_1\| \|P_2\| \|Q_1\| \|Q_2\|$ и

$$c_p = \frac{(P_1, P_2)}{\|P_1\| \|P_2\|}, \quad c_q = \frac{(Q_1, Q_2)}{\|Q_1\| \|Q_2\|}.$$

Из [1, Предложение 5.2] следует, что $\tilde{\lambda}_1 = \lambda_1/S$ и $\tilde{\lambda}_2 = \lambda_2/S$ являются собственными значениями матрицы

$$\mathbf{B} = \begin{pmatrix} pq + c_p c_q & qc_p + p^{-1} c_q \\ pc_q + q^{-1} c_p & p^{-1} q^{-1} + c_p c_q \end{pmatrix}.$$

Поэтому $\tilde{\lambda}_1$ и $\tilde{\lambda}_2$ могут быть найдены как корни многочлена

$$\tilde{\lambda}^2 - (pq + (pq)^{-1} + 2c_p c_q)\tilde{\lambda} + 1 + c_p^2 c_q^2 - c_p^2 - c_q^2 = 0.$$

Если $L\omega$ и $K\omega$ — целые числа, то $c_p = c_q = 0$, $\|P_1\|^2 = \|P_2\|^2 = L/2$ and $\|Q_1\|^2 = \|Q_2\|^2 = A^2K/2$.

Таким образом, мы имеем квадратное уравнение $\tilde{\lambda}^2 - 2\tilde{\lambda} + 1 = 0$ с одним кратным корнем $\tilde{\lambda}_1 = \tilde{\lambda}_2 = 1$. Отсюда, $\lambda_1 = \lambda_2 = A^2LK/4$.

В асимптотическом случае (N стремится к бесконечности) квадратное уравнение имеет вид $\tilde{\lambda}^2 - b(N)\tilde{\lambda} + c(N) = 0$, где коэффициенты $b(N) \rightarrow 2$ и $c(N) \rightarrow 1$ для $N \rightarrow \infty$. Так как S эквивалентна $A^2LK/4$, $\lambda_1 \rightarrow A^2LK/4$, $\lambda_2 \rightarrow A^2LK/4$.

2. Траекторная матрица $(S^{(1)}, S^{(2)})$ может быть выражена как $\mathbf{X} = P_1 Q_1^\top + P_2 Q_2^\top$, где P_1 и P_2 те же, что и в одномерном случае и

$$Q_1 = \begin{pmatrix} A \cos(2\pi\omega 0 + \varphi_1) \\ \vdots \\ A \cos(2\pi\omega(K-1) + \varphi_1) \\ B \cos(2\pi\omega 0 + \varphi_2) \\ \vdots \\ B \cos(2\pi\omega(K-1) + \varphi_2) \end{pmatrix}, Q_2 = \begin{pmatrix} A \sin(2\pi\omega 0 + \varphi_1) \\ \vdots \\ A \sin(2\pi\omega(K-1) + \varphi_1) \\ B \sin(2\pi\omega 0 + \varphi_2) \\ \vdots \\ B \sin(2\pi\omega(K-1) + \varphi_2) \end{pmatrix}.$$

Тогда $\|Q_1\|^2 = \|Q_2\|^2 = (A^2 + B^2)K/2$, если $L\omega$ и $K\omega$ целые числа, или стремится к этому числу в асимптотическом случае. Дальнейшие шаги доказательства те же, что и для одномерных временных рядов.

□

4.2.4. Комментарии по сравнению 1D-SSA и MSSA

Ковариационная структура

Рассмотрим более подробно случай двух временных рядов $X = (F, G)$ и пусть F и G — траекторные матрицы F и G соответственно. Тогда, поскольку в MSSA мы стыкуем отдельные траекторные матрицы горизонтально, траекторной матрицей X будет $X = [F : G]$. В соответствии с (1.12), SVD матрицы $X = X^{(H)}$ — это $X = \sum_i \sqrt{\lambda_i} U_i V_i^T$, где λ_i и U_i — собственные значения и собственные векторы матрицы $S = S_{MSSA}^{(H)} = XX^T = FF^T + GG^T$ и $V_i = X^T U_i / \sqrt{\lambda_i}$.

Рассмотрим теперь вертикальную укладку траекторных матриц F и G в траекторную матрицу: $X^{(V)} = \begin{pmatrix} F^T \\ G^T \end{pmatrix} = (X^{(H)})^T$. Сингулярное разложение матрицы $X^{(V)}$ тогда имеет вид $X^{(V)} = \sum_i \sqrt{\lambda_i} V_i U_i^T$, т.е. это транспонированное сингулярное разложение матрицы X . Здесь λ_i , V_i и U_i точно такие же, как и выше. но теперь они имеют другую интерпретацию: в частности, V_i (их часто называют EEOFs, см. пункт 5 замечания 35 в разделе 4.1.1) являются собственными векторами функции

$$S_{MSSA}^{(V)} = X^{(V)} (X^{(V)})^T = \begin{pmatrix} F^T F & F^T G \\ G^T F & G^T G \end{pmatrix}.$$

Последняя формула наглядно демонстрирует связь между двумя версиями (горизонтальная и вертикальная укладки) MSSA и показывает, что MSSA учитывает перекрестные ковариации временных рядов (точнее, мы получаем перекрестные ковариации, если на этапе предварительной обработки выполняется центрирование одномерных рядов).

Выбор длины окна

Выбор длины окна для одномерного SSA был рассмотрен в разделе 3.4. Проблема выбора длины окна в MSSA является более сложной, чем в 1D-SSA. До сих пор не существует глубокого исследования проблемы выбора оптимальной длины окна для анализа и, в еще большей степени, для прогнозирования многомерных временных рядов. Более того, выбор оптимальной длины окна для прогнозирования MSSA различается для различных типов методов прогнозирования; некоторые численные исследования этой проблемы проведены в разделе 4.3.3.

По аналогии с одномерным случаем мы можем сформулировать некоторые ключевые принципы выбора L . Основной принцип такой же, как и для 1D-SSA, и гласит, что выбор

L должен обеспечивать (приближенную) разделимость рядов. Однако случай MSSA имеет дополнительные особенности. Различные подходы к выбору длины окна можно частично объяснить следующим образом. В 1D-SSA имеет смысл ограничить длину окна интервалом $2 \leq L \leq [(N+1)/2]$, так как разложения SVD для длин окон $2 \leq L \leq [(N+1)/2]$ длиной L и $N-L+1$ совпадают. Для MSSA-анализа более чем одного временных рядов, разложения для всех возможных длин окон $2 \leq L \leq \min_i N_i - 1$, как правило, отличаются. В частности, если в 1D-SSA-анализе не имеет смысла брать $L > (N+1)/2$, в MSSA-анализе имеет смысл выбирать большие L (и, следовательно, малые $K_i = N_i - L + 1$) для выделения и сглаживания тренда.

Поскольку L в 1D-SSA не превышает половины длины временного ряда, то рекомендуется делимость $L = \min(L, K)$ на возможные периоды колебаний в 1D-SSA. В MSSA $\min(L, K_i)$ не обязательно равен L и поэтому также необходимо обращать внимание на значения K_i .

В 1D-SSA наиболее детальное разложение может быть получено, если траекторная матрица \mathbf{X} имеет максимальный ранг. В общем случае методов SSA-семейства, это соответствует случаю квадратной траекторной матрицы. Таким образом, для системы из s временных рядов длины N длина окна в MSSA, обеспечивающая квадратную траекторную матрицу \mathbf{X} составляет примерно $sN/(s+1)$. Для случая двух временных рядов это соответствует $2L/3$ для MSSA.

Численные исследования показывают, что формула $L \simeq sN/(s+1)$ подходит для разложения небольшого числа временных рядов (см. результаты моделирования в разделе 4.3.3), но не выглядит подходящей для системы многих коротких рядов (значения K_i становятся слишком малы для достижения разделимости). В целом, выбор $L \simeq N/2$ все еще подходит для MSSA.

Различные специальные методы могут быть перенесены из 1D-SSA в MSSA, например, Sequential SSA. Последовательный 1D-SSA основан на последовательном применении 1D-SSA с разной длиной окна, подробнее см. [29, Раздел 2.5.5]. Последовательный MSSA может быть применен аналогичным образом. В дополнение к причинам, которые аналогичны одномерному случаю, мы можем найти дополнительные аргументы в пользу Sequential MSSA. В частности, если тренды различных одномерных рядов имеют различную структуру, можно применить меньшую длину окна для достижения сходства собственных векторов и улучшить разделимость. После этого остатки с общей структурой (например, содержащие сезонность) могут быть одновременно разложены с большей длиной окна.

4.3. MSSA прогнозирование

Вспомним, следуя разделу 1.9.3, что прогнозирование в 1D-SSA выполняется для компоненты сигнала, которая может быть отделена с помощью 1D-SSA и управляет, возможно приближенно, ЛРС. Для краткости мы будем говорить о прогнозировании всего сигнала. 1D-SSA обеспечивает оценку подпространства сигнала и, таким образом, оценку одного из

ЛРС, управляющих сигналом. Рекуррентное прогнозирование 1D-SSA продолжает оцениваемый сигнал по оцененному ЛРС. Векторное 1D-SSA прогнозирование продолжает восстановленные векторы в заданном подпространстве. Методы одномерного SSA прогнозирования в заданном подпространстве описаны в разделе 1.9.3.

Как и в 1D-SSA, методы прогнозирования MSSA можно разделить на рекуррентное и векторное прогнозирование. В отличие от 1D-SSA, строки и столбцы траекторной матрицы в MSSA имеют разную структуру. Поэтому существует два вида MSSA-прогнозирования: по строкам и по столбцам; это зависит от того, в каком из двух пространств осуществляется прогнозирование (в пространстве строк или столбцов, соответственно). Всего существует четыре основных варианта MSSA-прогнозирования: рекуррентное прогнозирование по столбцам, рекуррентное прогнозирование по строкам, векторное прогнозирование по столбцам и векторное прогнозирование по строкам.

Существуют разные названия для одних и тех же методов прогнозирования. В [35], прогнозирование по столбцам и по строкам называется *L*- и *K*- прогнозированием. В [172], эти методы называются горизонтальным и вертикальным прогнозированием, а траекторная матрица транспонируется. В разделах 4.1.1 и 4.2.4 мы объяснили выбор ориентации траекторной матрицы MSSA и связь между горизонтально и вертикально сстыкованными траекторными матрицами отдельных временных рядов. Мы используем название ‘столбец’ и ‘строка’ по отношению к горизонтально сстыкованным траекторным матрицам, как определено в разделе 4.1.1. Вероятно, первая версия рекуррентного MSSA прогнозирования по строкам была предложена в [32].

В методах прогнозирования по столбцам каждый временной ряд в системе прогнозируется отдельно, но в заданном общем подпространстве (т.е. с использованием общего ЛРС). В методах прогнозирования по строкам каждый ряд прогнозируется с помощью собственного ЛРС, применяемого ко всему набору рядов системы. Опишем все четыре варианта прогнозирования MSSA.

4.3.1. Метод

Общие обозначения

Сначала мы введем некоторые общие обозначения, используемые для описания всех вариантов MSSA-прогнозирования.

Обозначим через $\bar{A} \in \mathbb{R}^{Q-1}$ векторы, состоящие из последних $Q - 1$ координат $A \in \mathbb{R}^Q$; то есть, векторы с удаленной первой координатой обозначаются линией в верхней части вектора. Обозначим через $\underline{A} \in \mathbb{R}^{Q-1}$ векторы, состоящие из первых $Q - 1$ координат A ; через $\pi(A)$ обозначим последнюю координату вектора. Для матрицы $\mathbf{A} = [A_1 : \dots : A_r]$ обозначим $\bar{\mathbf{A}} = [\bar{A}_1 : \dots : \bar{A}_r]$ и $\underline{\mathbf{A}} = [\underline{A}_1 : \dots : \underline{A}_r]$ и пусть $\boldsymbol{\pi}(\mathbf{A}) = (\pi(A_1), \dots, \pi(A_r))^T$ является последней строкой матрицы \mathbf{A} .

Рассмотрим следующую форму вектора $B \in \mathbb{R}^K$, где $K = \sum_{i=1}^s K_i$, обусловленную струк-

турой траекторного пространства ряда:

$$B = \begin{pmatrix} B^{(1)} \\ B^{(2)} \\ \vdots \\ B^{(s)} \end{pmatrix}, \quad \underline{\underline{B}} = \begin{pmatrix} \underline{B}^{(1)} \\ \underline{B}^{(2)} \\ \vdots \\ \underline{B}^{(s)} \end{pmatrix}, \quad \bar{\bar{B}} = \begin{pmatrix} \bar{B}^{(1)} \\ \bar{B}^{(2)} \\ \vdots \\ \bar{B}^{(s)} \end{pmatrix}, \quad (4.10)$$

где $B^{(j)} \in \mathbb{R}^{K_j}$, и пусть $\boldsymbol{\mu}(B) = (\pi(B^{(1)}), \dots, \pi(B^{(s)}))^T$. Также, для $\mathbf{B} = [B_1 : \dots : B_r]$ пусть $\underline{\underline{\mathbf{B}}} = [\underline{B}_1 : \dots : \underline{B}_r]$ и $\mathbf{B}^{(j)} = [B_1^{(j)} : \dots : B_r^{(j)}]$.

Предположим, что группа I , соответствующая прогнозируемой компоненте, задается множеством ведущих компонент на шаге разложения алгоритма 4.1; это предположение сделано только для упрощения формул. Итак, пусть r ведущих собственных троек $(\sqrt{\lambda_j}, U_j, V_j)$ выбраны как связанные с сигналом ранга r , поэтому $I = I_1 = \{1, \dots, r\}$, $\mathbf{U} = [U_1 : \dots : U_r]$, $\mathbf{V} = [V_1 : \dots : V_r]$. Восстановленный ряд $\tilde{\mathbf{X}}$, его траекторная матрица $\tilde{\mathbf{X}}$ и восстановленная матрица $\hat{\mathbf{X}}$ определены в разделе 1.2.1. Определим $\mathcal{L}^{\text{col}} = \text{span}(U_i, i \in I)$, $\mathcal{L}^{\text{row}} = \text{span}(V_i, i \in I)$. Восстановленная матрица $\hat{\mathbf{X}} = [\hat{\mathbf{X}}_1 : \dots : \hat{\mathbf{X}}_K]$ состоит из векторов-столбцов, которые являются проекциями векторов-столбцов траекторной матрицы на выбранное подпространство \mathcal{L}^{col} .

Чтобы не повторять знак транспонирования, обозначим $\tilde{\mathbf{Y}} = [\tilde{\mathbf{Y}}_1 : \dots : \tilde{\mathbf{Y}}_L] = \tilde{\mathbf{X}}^T$, $\hat{\mathbf{Y}} = [\hat{\mathbf{Y}}_1 : \dots : \hat{\mathbf{Y}}_L] = \hat{\mathbf{X}}^T$, $\hat{\mathbf{Y}}_k = \hat{\mathbf{X}}_k^T$.

Рекуррентный MSSA-прогноз

Обозначим вектор прогнозируемых значений сигнала для каждого временного ряда через $R_N = (\tilde{x}_{N_1+1}^{(1)}, \tilde{x}_{N_2+1}^{(2)}, \dots, \tilde{x}_{N_s+1}^{(s)})^T$. Рекуррентное прогнозирование тесно связано с заполнением пропусков в компонентах векторов из данного подпространства и фактически использует результат из предложения 14. Следуя [35], мы напишем формулы прогнозирования для двух версий рекуррентного прогноза MSSA: по столбцам (генерируемого $\{U_j\}_{j=1}^r$) и по строкам (генерируемого $\{V_j\}_{j=1}^r$). Эти формулы прогнозирования на один шаг вперед могут быть применены для M -шагов прогнозирования с помощью рекуррентного метода.

Рекуррентное прогнозирование по столбцам выполняет прогноз с помощью ЛРС порядка $L - 1$ применяемой к последним $L - 1$ точкам восстановленного сигнала; то есть, одно и то же ЛРС и разные начальные данные. Рекуррентное прогнозирование по строкам строит s различных линейных отношений, каждое из которых применяется к набору $K_i - 1$ последних точек ряда; то есть, ЛРС разные, но начальные данные для них одинаковые.

Прогноз по столбцам Обозначим через \mathbf{Z} матрицу, состоящую из последних $L - 1$ значений восстановленных сигналов:

$$\mathbf{Z} = \begin{pmatrix} \tilde{x}_{N_1-L+2}^{(1)} & \cdots & \tilde{x}_{N_1}^{(1)} \\ \tilde{x}_{N_2-L+2}^{(2)} & \cdots & \tilde{x}_{N_2}^{(2)} \\ \vdots & \vdots & \vdots \\ \tilde{x}_{N_s-L+2}^{(s)} & \cdots & \tilde{x}_{N_s}^{(s)} \end{pmatrix},$$

$v^2 = \sum_{j=1}^r \pi(U_j)^2$. Если $v^2 < 1$, то MSSA-прогноз по столбцам однозначно определен и может быть вычислен по формуле

$$R_N = \mathbf{Z}\mathcal{R}_L, \quad \text{где} \quad \mathcal{R}_L = \frac{1}{1-v^2} \sum_{j=1}^r \pi(U_j) \underline{U_j} \in \mathbb{R}^{L-1}. \quad (4.11)$$

Заметим, что (4.11) подразумевает, что прогнозирование всех отдельных сигналов осуществляется по одному и тому же ЛРС, который генерируется всей системой рядов.

Прогноз по строкам Введем векторы последних $K_m - 1$ значений восстановленных сигналов

$$\mathbf{Z}^{(m)} = (\tilde{x}_{N-K_m+2}^{(m)}, \dots, \tilde{x}_{N_m}^{(m)})^T, \quad m = 1, \dots, s,$$

и обозначим

$$\mathbf{Z} = \begin{pmatrix} \mathbf{Z}^{(1)} \\ \mathbf{Z}^{(2)} \\ \vdots \\ \mathbf{Z}^{(s)} \end{pmatrix}, \quad \mathbf{S} = [\boldsymbol{\mu}(V_1) : \dots : \boldsymbol{\mu}(V_r)].$$

В этих обозначениях $Z = \bar{\bar{\bar{Y}}}_L$.

Если обратная матрица $(\mathbf{I}_s - \mathbf{S}\mathbf{S}^T)^{-1}$ существует и $r \leq K - s$, то рекуррентный MSSA-прогноз по строкам существует и может быть вычислен по формуле

$$R_N = \mathcal{R}_K Z, \quad \text{где} \quad \mathcal{R}_K = (\mathbf{I}_s - \mathbf{S}\mathbf{S}^T)^{-1} \mathbf{S}\underline{\mathbf{V}}^T. \quad (4.12)$$

Заметим, что (4.12) подразумевает, что прогнозирование отдельных сигналов осуществляется с помощью ЛРС, которые различны для разных рядов. Значение прогноза в общем случае зависит от последних значений всех временных рядов из системы временных рядов.

Векторный MSSA-прогноз

Обозначим $\underline{\mathcal{L}}^{\text{col}} = \text{span}(\underline{U}_1, \dots, \underline{U}_r)$ и $\underline{\mathcal{L}}^{\text{row}} = \text{span}(\underline{V}_1, \dots, \underline{V}_r)$. Пусть Π^{col} — ортогональный проектор векторов из \mathbb{R}^{L-1} на $\underline{\mathcal{L}}^{\text{col}}$ и Π^{row} — ортогональный проектор \mathbb{R}^{K-s} на $\underline{\mathcal{L}}^{\text{row}}$.

Явную форму матриц проекторов столбцов и строк дана в предложении 15. Однако, вычисления по этой формуле занимают много времени. Быстрый алгоритм расчета представлен в разделе 4.3.2.

Прогноз по столбцам Мы упоминали выше, что для заданного подпространства ($\underline{\mathcal{L}}^{\text{col}}$ в нашем случае) прогноз по столбцам выполняется независимо для каждого временного ряда. Определим линейный оператор $\mathcal{P}_{\text{Vec}}^{\text{col}} : \mathbb{R}^L \mapsto \underline{\mathcal{L}}^{\text{col}}$ по формуле

$$\mathcal{P}_{\text{Vec}}^{\text{col}} Z = \begin{pmatrix} \Pi^{\text{col}} \bar{Z} \\ \mathcal{R}_L^T \bar{Z} \end{pmatrix}, \quad (4.13)$$

где \mathcal{R}_L определено в (4.11).

Алгоритм *векторного прогнозирования* для j -го ряда выглядит следующим образом.

1. В приведенных выше обозначениях определим векторы Z_i следующим образом:

$$Z_i = \begin{cases} \hat{X}_i^{(j)} & \text{для } i = 1, \dots, K_j, \\ \mathcal{P}_{\text{Vec}}^{\text{col}} Z_{i-1} & \text{для } i = K_j + 1, \dots, K_j + M + L - 1. \end{cases} \quad (4.14)$$

2. Построив матрицу $\mathbf{Z} = [Z_1 : \dots : Z_{K_j+M+L-1}]$ и сделав ее диагональное усреднение, мы получаем ряд $z_1, \dots, z_{N_j+M+L-1}$.
3. Числа $z_{N_j+1}, \dots, z_{N_j+M}$ образуют M членов векторного прогноза.

Прогноз по строкам Определим линейный оператор $\mathcal{P}_{\text{Vec}}^{\text{row}} : \mathbb{R}^K \mapsto \underline{\mathcal{L}}^{\text{row}}$ по формуле

$$\mathcal{P}_{\text{Vec}}^{\text{row}} Z = A, \quad (4.15)$$

при этом $\underline{A} = \Pi^{\text{row}} \bar{Z}$ и $\boldsymbol{\mu}(A) = \mathcal{R}_K \bar{Z}$, где \mathcal{R}_K определено в (4.12).

Алгоритм *векторного прогнозирования* состоит в следующем.

1. В приведенных выше обозначениях, определим векторы Z_i следующим образом:

$$Z_i = \begin{cases} \hat{Y}_i & \text{for } i = 1, \dots, L, \\ \mathcal{P}_{\text{Vec}}^{\text{row}} Z_{i-1} & \text{for } i = L + 1, \dots, L + M + K^* - 1, \end{cases} \quad (4.16)$$

где $K^* = \max(K_i, i = 1, \dots, s)$.

2. Построив матрицу $\mathbf{Z} = [Z_1 : \dots : Z_{L+M+K^*-1}]$ и сделав шаг восстановления, мы получаем ряд $z_1^{(j)}, \dots, z_{N_j+M+K^*-1}^{(j)}$, $j = 1, \dots, s$.

3. Числа $z_{N_j+1}^{(j)}, \dots, z_{N_j+M}^{(j)}$, $j = 1, \dots, s$, образуют M членов векторного прогноза.

Замечание 36. Для векторного прогноза на M шагов вперед строится $M+K^*-1$ новых запаздывающих векторов для прогнозирования по строкам и $M+L-1$ для прогнозирования по столбцам. Причина в том, что прогноз на M шагов наследует прогноз на $(M-1)$ шагов как его часть. Эта особенность векторного прогнозирования обеспечивает его устойчивость и точность, если прогнозируется точно извлеченная компонента конечного ранга, т.е. когда долгосрочный прогноз имеет смысл. В противном случае (если аппроксимация, построенная по MSSA, неадекватна), долгосрочное векторное прогнозирование может привести к заблуждению, и даже краткосрочное векторное прогнозирование может быть неточным при больших K^* или L соответственно.

4.3.2. Быстрый алгоритм векторного прогнозирования

В [29, стр. 76] упоминается, что векторное прогнозирование занимает много времени, в то время как рекуррентное прогнозирование является быстрым. Однако, это так, если реализовать алгоритм 1.9 непосредственно. Оказывается, что можно значительно ускорить векторное прогнозирование. Более того, в текущей реализации в RSSA векторное прогнозирование даже немного быстрее, чем рекуррентное. Опишем полученный результат, следуя [12, Раздел 6.3].

Обозначим через \dagger псевдоинверсию матрицы.

MSSA прогноз по столбцам

Как было показано в разделе 4.3.1, векторное MSSA-прогнозирование по столбцам сводится к выполнению s одномерных векторных прогнозов в том же подпространстве \mathcal{L}^{col} . Далее мы описываем алгоритм для быстрого векторного прогнозирования одномерных временных рядов в заданном подпространстве \mathcal{L}^{col} . (т.е. мы описываем быструю версию векторного прогнозирования в Basic SSA, см. алгоритм 1.9).

Рассмотрим прогнозирование в подпространстве \mathcal{L}^{col} , заданном базисом $\{P_1, \dots, P_r\}$. Обозначим $\mathbf{P} = [P_1 : \dots : P_r]$. Каждый восстановленный вектор \widehat{X}_k одномерного временного ряда принадлежит \mathcal{L}^{col} ; следовательно, существуют коэффициенты $W_k \in \mathbb{R}^r$ такие, что $\widehat{X}_k = \mathbf{P}W_k$. Обозначим $\mathbf{W} = [W_1 : \dots : W_K]$. Фактически, входными данными для алгоритма является минимальное разложение $\widehat{\mathbf{X}}$ в сумму элементарных матриц ранга 1 в виде $\widehat{\mathbf{X}} = \mathbf{P}\mathbf{Q}^\top$ и $\mathbf{W} = \mathbf{Q}^\top$.

Заметим, что если мы имеем сингулярное разложение матрицы $\widehat{\mathbf{X}} = [\widehat{X}_1 : \dots : \widehat{X}_K]$, то левые сингулярные векторы обеспечивают базис подпространства, а W_k определяются правыми сингулярными векторами и сингулярными значениями: $\mathbf{P} = [U_1 : \dots : U_r]$ и $\mathbf{Q} = [\sqrt{\lambda}_1 V_1 : \dots : \sqrt{\lambda}_r V_r]$.

В векторном прогнозировании мы расширяем восстановленную матрицу как $\mathbf{Z} = [\widehat{\mathbf{X}} : Z_{K+1} : \dots : Z_{K+M+L-1}]$, где векторы Z_k , $k \geq K+1$, получаются как $Z_k = \mathcal{P}_{\text{Vec}}^{\text{col}} Z_{k-1}$ (и $Z_k = \widehat{X}_k$ для

$k = 1, \dots, K$). Так как все Z_k принадлежат \mathcal{L}^{col} , то существуют $W_k \in \mathbb{R}^r$, $k \geq K+1$ такие, что

$$\mathbf{Z} = \mathbf{P}[W_1 : \dots : W_{K+M+L-1}].$$

Таким образом, существует матрица \mathbf{D} такая, что $W_k = \mathbf{D}W_{k-1}$ для $k \geq K+1$. Это наблюдение приводит к следующему алгоритму для векторного прогнозирования.

АЛГОРИТМ 4.3: Быстрое векторное прогнозирование по столбцам

Input: \mathbf{P}, \mathbf{W} .

Output: прогнозируемые значения z_{N+1}, \dots, z_{N+M} .

- 1: Вычислить матрицу $\mathbf{D} = \underline{\mathbf{P}}^\dagger \bar{\mathbf{P}}$ используя QR-разложение [173].
- 2: Для $k = K+1, \dots, K+M+L-1$ вычислить $W_k = \mathbf{D}W_{k-1}$. Применить быстрый алгоритм ганкелизации матрицы ранга один [12, Алгоритм 2] для вычисления матрицы $\mathbf{Z} = \mathbf{P}[\mathbf{W} : W_{K+1} : \dots : W_{K+M+L-1}]$, которая в явном виде выражается как сумма матриц ранга один, и получить ряд $z_1, \dots, z_{N+M+L-1}$.
- 3: Числа z_1, \dots, z_{N+M} образуют обновленные восстановленный и прогнозируемый ряды.

Чтобы доказать правильность алгоритма, остается показать, что формула $\mathbf{D} = \underline{\mathbf{P}}^\dagger \bar{\mathbf{P}}$ верна, что будет рассмотрено в конце этого раздела. Наконец, есть еще одно улучшение в вычислении \mathbf{D} .

Замечание 37. 1. Пункт 1 алгоритма 4.3 строит матрицу сдвига из метода LS-ESPRIT для оценки частоты, кратко описанного в разделе 1.9.2.

2. Если $\{P_i\}$ является ортонормированный системой, то \mathbf{D} может быть вычислена без QR-разложения как:

$$\mathbf{D} = \left(\mathbf{I}_r - \frac{1}{1 - \boldsymbol{\pi}^\top \boldsymbol{\pi}} \boldsymbol{\pi} \boldsymbol{\pi}^\top \right) \underline{\mathbf{P}}^\top \bar{\mathbf{P}},$$

где $\boldsymbol{\pi} = \boldsymbol{\pi}(\mathbf{P})$, \mathbf{I}_r — единичная матрица $r \times r$.

MSSA прогноз по строкам

Векторный прогноз по строкам несколько отличается от векторного прогноза по столбцам, но идея та же самая. Мы имеем дело с прогнозированием в подпространстве строк $\mathcal{L}^{\text{row}} = \text{span}(Q_1, \dots, Q_r)$ и продолжаем последовательность векторов-строк \hat{Y}_k из $\hat{\mathbf{X}}$. Если векторы-строки равны $\hat{Y}_k = \mathbf{Q}W_k$, $k = 1, \dots, L$, то $\hat{\mathbf{X}}^\top = \mathbf{Q}\mathbf{W}$, где $\mathbf{W} = [W_1 : \dots : W_L]$. В следующем алгоритме, как и в прогнозировании по столбцам, векторы W_k продолжаются вместо векторов Y_k .

 АЛГОРИТМ 4.4: Быстрое прогнозирование вектора строки

Input: \mathbf{Q}, \mathbf{W} .

Output: прогнозируемые значения $z_{N_p+1}^{(p)}, \dots, z_{N_p+M}^{(p)}$, $p = 1, \dots, s$.

- 1: Вычислить матрицу $\mathbf{D} = \underline{\underline{\mathbf{Q}}}^\dagger \overline{\overline{\mathbf{Q}}}$ с помощью QR-разложения.
 - 2: Для $k = L + 1, \dots, L + M + \max_{p=1,\dots,s} K_p - 1$ вычислить $W_k = \mathbf{D}W_{k-1}$.
 - 3: Применить быстрый алгоритм ганкелизации матрицы ранга один [12, Алгоритм 2] для вычисления каждой из s матриц $\mathbf{Z}^{(p)} = \mathbf{Q}^{(p)}[\mathbf{W} : W_{L+1} : \dots : W_{L+M+K_p-1}]$, $p = 1, \dots, s$, и получить s рядов $z_1^{(p)}, \dots, z_{N_p+M+L-1}^{(p)}$.
 - 4: Числа $z_1^{(p)}, \dots, z_{N_p+M}^{(p)}$ образуют обновленные восстановленный и прогнозируемый ряды.
-

Замечание 38. Если $\{Q_i\}$ — ортонормированная система, то \mathbf{D} из алгоритма 4.4 можно выразить как

$$\mathbf{D} = (\mathbf{I}_r - \mathbf{S}^\top (\mathbf{I}_s - \mathbf{S}\mathbf{S}^\top)^{-1} \mathbf{S}) \underline{\underline{\mathbf{Q}}}^\top \overline{\overline{\mathbf{Q}}},$$

где $\mathbf{S} = [\boldsymbol{\mu}(Q_1) : \dots : \boldsymbol{\mu}(Q_r)]$, \mathbf{I}_r и \mathbf{I}_s — это единичные матрицы размеров $r \times r$ и $s \times s$.

Доказательство корректности алгоритмов

Для простоты мы рассмотрим одномерный случай, то есть векторный SSA-прогноз, который для одномерных рядов совпадает со MSSA-прогнозом по столбцам. Доказательство для прогнозирования по строкам аналогично.

Нам нужно доказать, что $Z_{k+1} = \mathbf{P}W_{k+1}$, где $W_{k+1} = \mathbf{D}W_k$, $\mathbf{D} = \underline{\underline{\mathbf{P}}}^\dagger \overline{\overline{\mathbf{P}}}$. Достаточно доказать, что $Z_{k+1} = \underline{\underline{\mathbf{P}}}W_{k+1}$, так как последняя координата вектора однозначно определена. В стандартной формулировке алгоритма векторного прогнозирования, приведенной в алгоритме 1.9, Z_{k+1} является проекцией на пространство столбцов $\underline{\underline{\mathbf{P}}}$, т.е., $Z_{k+1} = \underline{\underline{\mathbf{P}}}\underline{\underline{\mathbf{P}}}^\dagger \overline{\overline{\mathbf{Z}}}_k$. Так как $\overline{\overline{\mathbf{P}}}W_k$ — это в точности $\overline{\overline{\mathbf{Z}}}_k$ по определению W_k , имеем, что $Z_{k+1} = \underline{\underline{\mathbf{P}}}(\underline{\underline{\mathbf{P}}}^\dagger \overline{\overline{\mathbf{P}}})W_k = \underline{\underline{\mathbf{P}}}W_{k+1}$; тем самым, эквивалентность стандартного и быстрого алгоритмов векторного прогнозирования доказана.

Замечание 39. Ускорение реализации алгоритмов объясняется двумя причинами:

1. умножение на матрицы малого размера $r \times r$ на каждом шаге вместо умножения на матрицы гораздо большего размера,
2. форма матрицы, которую нужно ганкелизовать, является подходящей для применения быстрого алгоритма ганкелизации матрицы ранга 1.

4.3.3. Пример моделирования: численное сравнение

В этом разделе мы демонстрируем, как точность MSSA связана со структурой многомерного временного ряда. Цель состоит в том, чтобы сравнить точность для раздельного

анализа и прогнозирования временных рядов с точностью при использовании одновременной обработки системы рядов.

В приведенном ниже исследовании мы рассматриваем случай $s = 2$ и исследуем следующие методы SSA: (a) 1D-SSA, применяемый дважды, и (b) MSSA. Исследуемые примеры моделей включают наименее благоприятные и наиболее благоприятные случаи для MSSA.

Предположим, что мы наблюдаем $(X^{(1)}, X^{(2)}) = (H^{(1)}, H^{(2)}) + (N^{(1)}, N^{(2)})$, где $(H^{(1)}, H^{(2)})$ — это двумерный сигнал, состоящий из двух гармонических временных рядов, $N^{(1)}$ и $N^{(2)}$ являются реализациями независимых белых гауссовских шумов. Затем мы можем использовать стандартные методы моделирования для получения оценок среднеквадратичных ошибок (MSE) для восстановления и прогнозирования $(H^{(1)}, H^{(2)})$ указанными методами SSA. Результатирующая MSE рассчитывается как среднее из $MSE^{(1)}$ и $MSE^{(2)}$ для $H^{(1)}$ и $H^{(2)}$ соответственно.

Для моделирования временного ряда мы берем следующие параметры: $N = 71$, дисперсия всех компонентов шума $\sigma^2 = 25$, число повторений 10000. Мы рассматриваем следующие две версии сигнала $(H^{(1)}, H^{(2)})$.

Пример А (одинаковые периоды):

$$h_k^{(1)} = 30 \cos(2\pi k/12), \quad h_k^{(2)} = 20 \cos(2\pi k/12 + \pi/4), \quad k = 1, \dots, N.$$

Пример С (разные периоды):

$$h_k^{(1)} = 30 \cos(2\pi k/12), \quad h_k^{(2)} = 20 \cos(2\pi k/8 + \pi/4), \quad k = 1, \dots, N.$$

Выбор этих примеров обусловлен тем, что размеры траекторного пространства сигналов (т.е. рангов) различны для различных расширений метода 1D-SSA, см. таблицу 4.1. Для каждого примера ранг выделенный синим цветом, соответствует методу с наилучшей точностью для данного примера. Ячейки в строке, соответствующей 1D-SSA, содержат одно число, так как ранги временных рядов из рассматриваемых наборов совпадают. Оценка проводилась по 10000 повторений.

Таблица 4.1: Размерность траекторного пространства сигнала.

	Пример А	Пример С
MSSA	2	4
1D-SSA	2	2

Результаты исследования для различных длин окон L обобщены в таблицах 4.2 и 4.3. Был выполнен прогноз на 24 шагов вперед. Для каждого примера ячейки, соответствующие методу с точностью восстановления/прогноза, которая близка к наилучшей, выделены жирным шрифтом, а общий минимум — синим цветом.

Таблица 4.2: MSE восстановления сигнала.

Пример А	$L = 12$	$L = 24$	$L = 36$	$L = 48$	$L = 60$
MSSA	3.18	1.83	1.59	1.47	2.00
1D-SSA	3.25	2.01	2.00	2.01	3.25
Пример С	$L = 12$	$L = 24$	$L = 36$	$L = 48$	$L = 60$
MSSA	6,91	3,77	3,07	2,88	3,84
1D-SSA	3.23	2.01	2.00	2.01	3.23

Таблица 4.3: MSE прогноза сигнала.

Пример А	$L = 12$	$L = 24$	$L = 36$	$L = 48$	$L = 60$
Рекуррентный					
MSSA-column	5.36	3.67	3.73	3.70	4.43
MSSA-row	6.02	4.25	3.83	3.32	3.98
1D-SSA	7.24	5.59	6.30	6.42	7.93
Векторный					
MSSA-column	5.93	3.77	3.62	3.11	3.65
MSSA-row	4.00	3.03	3.39	3.17	4.24
1D-SSA	7.74	5.43	5.85	5.14	6.76
Пример С	$L = 12$	$L = 24$	$L = 36$	$L = 48$	$L = 60$
Рекуррентный					
MSSA-column	25,76	7,39	7,55	7,43	9,00
MSSA-row	19.82	8.47	8.00	6.66	8.30
1D-SSA	7.36	5.61	6.28	6.44	8.00
Векторный					
MSSA-column	25.34	7.56	7.57	6.20	7.67
MSSA-row	57.59	6.04	7.03	6.30	8.69
1D-SSA	7.84	5.47	5.84	5.18	6.88

Сравнение таблиц 4.2 и 4.3 с таблицей 4.1 ясно показывает связь между точностью восстановления/прогноза сигнала и размерностью траекторного пространства сигнала.

Отметим, что восстановления по 1D-SSA совпадают для длин окон L и $N - L + 1$ (12 и 60, 24 и 48 для рассмотренных примеров). Восстановления по MSSA различны для разных L . Также отметим, что траекторная матрица для 1D-SSA имеет ранг $\min(L, N - L + 1)$ и ранг максимальен для $L \simeq N/2$. Траекторная матрица MSSA имеет ранг, равный $\min(L, (N - L + 1)s)$, где s число временных рядов в системе. Этот ранг максимальен для $L \simeq sN/(s + 1)$. Хотя максимальность ранга не гарантирует минимальность ошибок, это соображение означает, что для достижения лучшей разделимости выбор длины окна L больше, чем $N/2$, часто может быть рекомендован. Моделирование подтверждает это: минимум ошибки восстановления для MSSA достигается при $L = 48 = 72 \times 2/3$.

Ошибки прогнозирования имеют гораздо более сложную структуру, см. раздел 3.4. В частности, эти ошибки прогнозирования зависят от ошибки восстановления для последних

точек временного ряда; поэтому ошибка может иметь зависимость от L , которая отличается от зависимости для средних ошибок восстановления. Рассмотренные примеры показывают, что векторный прогноз является более точным, чем рекуррентный, и что MSSA прогноз по строкам немного точнее, чем прогноз по столбцам.

Рассмотренные примеры подтверждают следующие утверждения:

- Точность методов, основанных на SSA, тесно связана со структурой траекторных пространств сигналов, генерируемых этими методами. MSSA имеет преимущество, если временные ряды из системы имеют согласованные компоненты. (Обратите внимание, что мы рассматривали равные уровни шума).
- Оптимальная длина окна для анализа и прогнозирования может отличаться. Несмотря на то, что точность прогноза связана с точностью восстановления, эта связь не является однозначной.
- Векторный прогноз с наилучшей длиной окна более точен, чем рекуррентный прогноз. Однако это не всегда так, если сравнивать точность прогноза для одной и той же длины окна. Это, вероятно, справедливо для прогнозирования только хорошо отделенного сигнала конечного ранга, см. замечание 36.
- В MSSA рекомендации по выбору длины окна (например, “взять L больше (или меньше) чем половина длины временного ряда”) для рекуррентного прогнозирования в некотором смысле противоположны тем, что используются для векторного прогнозирования.
- Для прогнозирования по строкам и столбцам (методы прогнозирования 1D-SSA являются частным случаем прогнозирования по столбцам), рекомендации также являются противоположными. Это неудивительно, так как L и K поменялись местами в MSSA по сравнению с 1D-SSA.

4.4. Автоматизация группировки в MSSA

Для MSSA мы предлагаем обобщение низкочастотного метода из раздела 2.5.1, частотного метода, представленного в разделе 2.5.2, и метода регулярности углов, описанного в разделе 2.5.3 для идентификации колебательной компоненты [129].

Рассмотрим многомерный временной ряд $\mathbf{X} = \left(\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(s)} \right)$, $\mathbf{X}^{(p)} = \left(x^{(p)}(1), \dots, x^{(p)}(N_p) \right)$, $p = 1, \dots, s$, $x^{(p)}(i) \in \mathbb{R}$.

Главной особенностью MSSA, в отличие от вещественного или комплексного случая, является то, что все три вида объектов: элементарные восстановленные ряды, левые сингулярные векторы и правые сингулярные векторы, имеют различную структуру.

Из описания алгоритма MSSA следует, что восстановленные ряды имеют ту же форму, что и исходный многомерный ряд, т.е. они также являются многомерными рядами (или, что то же самое, системой временных рядов).

Из предложения 21 для MSSA следует, что левые сингулярные векторы являются одномерными вещественными временными рядами длины L . Они описывают общую структуру ряда $X^{(1)}, \dots, X^{(s)}$ (структуре пространства столбцов траекторной матрицы). Они имеют ту же форму, что и сингулярные векторы для единичного временного ряда.

Правые сингулярные векторы также являются одномерными вещественными временными рядами; они описывают структуру пространства строк траекторной матрицы; каждая строка состоит из элементов всех рядов $X^{(1)}, \dots, X^{(s)}$, а не только одного. Будем использовать разбиение правых сингулярных векторов V_i на части $V_i^{(m)}$, $m = 1, \dots, s$, введенное в (4.6).

Таким образом, алгоритмы автоматизации MSSA имеют различные модификации для различных вариантов входных объектов. Различные модификации алгоритмов могут давать различные результаты.

Метод низких частот для идентификации тренда

Давайте обобщим алгоритм 2.13 идентификации тренда на случай многомерных рядов. Алгоритм 2.13 может быть применен в 1D-SSA либо к левому или правому сингулярному вектору, либо к элементарному восстановленному ряду тем же способом. Как уже упоминалось, в MSSA левые сингулярные векторы являются одномерными вещественно-значными временными рядами длины L и имеют ту же форму, что и сингулярные векторы для одного ряда. То есть, левые сингулярные векторы имеют ту же форму, что и в случае 1D-SSA, и алгоритм 2.13 низкочастотного метода может быть применен к ним таким же образом. Алгоритм 4.5 для этого случая приведен ниже.

Правые сингулярные векторы являются одномерными вещественными временными рядами длины K и представлены в виде (4.6). Мы можем применить низкочастотный метод к каждой части $V_i^{(1)}, \dots, V_i^{(s)}$ правого сингулярного вектора V_i . Рассмотрим меру (2.20) для каждой части $V_i^{(1)}, \dots, V_i^{(s)}$ вектора V_i , и возьмем максимум. Предлагаемый алгоритм 4.6 формально описан ниже.

Элементарные восстановленные ряды в случае MSSA являются многомерными временными рядами. Идея применения к ним низкочастотного метода такая же, как и для правых сингулярных векторов: на первом шаге для каждой из s рядов считаем меру по формуле (2.20) и берем максимальное значение этой меры для использования на втором шаге. Этот вариант метода также задается алгоритмом 4.6, разница лишь в том, что в качестве входных данных берутся элементарные восстановленные ряды.

АЛГОРИТМ 4.5: MSSA. Низкочастотный метод для определения тренда: версия с левыми сингулярными векторами

Input: Следующие данные и параметры являются входными.

1. **Данные:** индексная группа $I \in \{1, \dots, d\}$; частота $0 \leq \omega \leq 0.5$; левые сингулярные векторы U_i , $i \in I$.

2. **Параметры:** порог $0 \leq T_0 \leq 1$.

Output: Группа индексов $J \subset I$ компонентов, связанных с трендом.

- 1: Для каждого вектора U_i , $i \in I$, мы вычисляем значение $T(U_i; \omega)$, используя (2.20).
 - 2: J — это группа индексов $i \in I$ таких, что $T(U_i; \omega) \geq T_0$.
-

АЛГОРИТМ 4.6: MSSA. Низкочастотный метод для тренда: версия с правыми сингулярными векторами или элементарный восстановленный ряд

Input: Следующие данные и параметры являются входными.

1. **Данные:** индексная группа $I \in \{1, \dots, d\}$; частота $0 \leq \omega \leq 0.5$; ряд Y_i , $i \in I$: правые сингулярные векторы или элементарные восстановленные ряды.
2. **Параметры:** порог $0 \leq T_0 \leq 1$.

Output: Группа индексов $J \subset I$ компонентов, связанных с трендом.

- 1: Для каждого ряда $Y_i = (Y_i^{(1)}, \dots, Y_i^{(s)})$, $i \in I$, вычислить значения $T(Y_i^{(1)}; \omega), \dots, T(Y_i^{(s)}; \omega)$, используя (2.20). $T_m := \max\{T(Y_i^{(1)}; \omega); \dots; T(Y_i^{(s)}; \omega)\}$.
 - 2: J — это группа индексов $i \in I$ таких, что $T_m \geq T_0$.
-

Частотный метод идентификации колебательной компоненты

В многомерном случае мы также будем считать, что осциллирующая компонента каждого ряда является суммой э.-м. гармоник, т.е. элементы многомерного э.-м. гармонического ряда S имеют вид (4.9).

В MSSA левые сингулярные векторы являются одномерными вещественными временными рядами длины L и имеют ту же форму, что и сингулярные векторы для одного ряда (см. раздел 4.2). Поэтому частотный метод из раздела 2.5.2 может быть применен к ним точно так же, как и в случае 1D-SSA. Ниже приведен алгоритм 4.7 частотного метода для MSSA в применении к левым сингулярным векторам.

АЛГОРИТМ 4.7: MSSA. Частотный метод для колебательной компоненты: версия с левыми сингулярными векторами

Input: Следующие данные и параметры являются входными.

1. **Данные:** левые сингулярные векторы $\{U_i\}_{i=1}^d$.
2. **Параметры:** параметр $s_0 \in \mathbb{Z}_+$, порог $\rho_0 \in [0, 1]$.

Output: Группа индексов компонент $I^{(P)}$, связанных с осциллирующей компонентой.

- 1: На основе $\{U_i\}_{i=1}^d$, получить индексную группу $J_1^{(P)}$ используя (2.21) с s_0 и индексную группу $J_2^{(P)}$ используя (2.22) с s_0 .
 - 2: На основе $\{U_j\}_{j \in J_1^{(P)} \cup J_2^{(P)}}$ получить индексную группу $J^{(P)}$, используя (2.25) с ρ_0 .
-

Как упоминалось выше, правые сингулярные векторы V_i имеют вид (4.6). Поскольку векторы, к которым мы применяем частотный метод, должны иметь норму 1, вместе с векторами $V_i^{(p)}$ мы рассмотрим их нормированные версии $\widehat{V}_i^{(p)} := \frac{V_i^{(p)}}{\|V_i^{(p)}\|}$.

На первом шаге метода вычисляется множество индексов $J_{1,p}^{(P)}$, $p = 1, \dots, s$:

$$J_{1,p}^{(P)} = \{i : \theta_i^{(p)}, \theta_{i+1}^{(p)} > 0, L|\theta_i^{(p)} - \theta_{i+1}^{(p)}| \leq s_0, 1 \leq i \leq d-1\}, \quad (4.17)$$

где $\theta_i^{(p)} = \arg \max_{0 < k \leq K_p/2} \{\Pi_{\widehat{V}_i^{(p)}}^{K_p}(k/K_p)\}$, $p = 1, \dots, s$. Таким образом, для каждого $p = 1, \dots, s$ множества $J_{1,p}^{(P)}$ вычисляются для каждой пары последовательных векторов $\widehat{V}_i^{(p)}$ и $\widehat{V}_{i+1}^{(p)}$ длины K_p .

Аналогично, каждый сингулярный вектор проверяется на гармонику с частотой 0.5:

$$J_{2,p}^{(P)} = \{i : K_p|\theta_i^{(p)} - 0.5| \leq s_0, 1 \leq i \leq d\}. \quad (4.18)$$

На втором шаге мы объединяем множества индексов: $J_1^{(P)}$ — объединение множеств $\{J_{1,p}^{(P)}\}_{p=1}^s$, $J_2^{(P)}$ — объединение множеств $\{J_{2,p}^{(P)}\}_{p=1}^s$.

Для второго шага мы вводим определение

$$\rho_{i,j} := \max_p \left\{ \max_{0 < k \leq K_p/2} \left(\rho_{\{\widehat{V}_i^{(p)}, \widehat{V}_j^{(p)}\}}(k/K_p) + \rho_{\{\{\widehat{V}_i^{(p)}, \widehat{V}_j^{(p)}\}}((k+1)/K_p) \right), p = 1, \dots, s \right\},$$

где $\widehat{V}_i^{(p)}$ и $\widehat{V}_j^{(p)}$ — векторы с индексами из множества $J_1^{(P)}$, построенного на первом шаге. Определение 9 меры ρ_A дано в разделе 2.5.2.

Для идентификации компонентов, принадлежащих гармонике с частотой 0.5, мера имеет вид

$$\rho_i := \max_p \left\{ \max_{0 \leq k \leq K_p/2} \left(\rho_{\{\widehat{V}_i^{(p)}\}}(\lfloor K_p/2 \rfloor / K_p) + \rho_{\{\{\widehat{V}_i^{(p)}\}}((\lfloor K_p/2 \rfloor + 1) / K_p) \right), p = 1, \dots, s \right\},$$

где $i \in J_2^{(P)}$.

Конечным результатом, см. алгоритм 4.8, являются индексы

$$J^{(P)} = \{(i, j) : \rho_{i,j} \geq \rho_0\} \cup \{i : \rho_i \geq \rho_0\}. \quad (4.19)$$

АЛГОРИТМ 4.8: MSSA. Частотный метод для колебательной компоненты: версия с правыми сингулярными векторами

Input: Следующие данные и параметры являются входными.

1. **Данные:** правые сингулярные векторы $\{V_i\}_{i=1}^d$.

2. **Параметры:** параметр $s_0 \in \mathbb{Z}_+$, порог $\rho_0 \in [0, 1]$.

Output: Группа индексов $J^{(P)}$ компонент, связанных с осциллирующей компонентой.

- 1: На основе $\{V_i\}_{i=1}^d$, получить группу индексов $J_{1,p}^{(P)}$, $p = 1, \dots, s$, используя (4.17) с s_0 и индексную группу $J_{2,p}^{(P)}$, $p = 1, \dots, s$, используя (4.18) с s_0 ; $J_1^{(P)}$ — объединение множеств $\{J_{1,p}^{(P)}\}_{p=1}^s$, $J_2^{(P)}$ — объединение множеств $\{J_{2,p}^{(P)}\}_{p=1}^s$.
 - 2: Индексная группа $J^{(P)}$ по $\{V_j\}_{j \in J_1^{(P)} \cup J_2^{(P)}}$ получена с помощью (4.19) с ρ_0 .
-

Метод идентификации колебательной компоненты по регулярности углов

Опять же, если мы рассматриваем левые сингулярные векторы, алгоритм может быть применен точно так же, как и в случае 1D-SSA (раздел 2.5.3). Алгоритм 4.9 для этого случая приведен ниже.

АЛГОРИТМ 4.9: MSSA. Метод идентификации колебательной компоненты по регулярности углов: по левым сингулярным векторам

Input: Следующие данные и параметры являются входными.

1. **Данные:** число компонент r ; левые сингулярные векторы $\{U_j\}_{j=1}^r$.

2. **Параметры:** число э.-м. гармоник $m \leq r/2$, или порог $t_0 \geq 0$, в зависимости от критерия остановки.

Output: Группа сингулярных векторных индексов J , связанных с колебательной компонентой.

- 1: На основе $\{U_i\}_{i=1}^r$, вычислить значения $\tilde{\tau}(U_j, U_{j+1})$, $j = 1, \dots, r-1$ используя (2.27) и упорядочить их по возрастанию. При вычислении для каждого $j = 2, \dots, r-1$ проверяем, если $\tilde{\tau}(U_j, U_{j+1}) < \tilde{\tau}(U_{j-1}, U_j)$, то отбрасываем $\tilde{\tau}(U_{j-1}, U_j)$ из рассмотрения, иначе отбрасываем $\tilde{\tau}(U_j, U_{j+1})$. В результате получаем значения $\tau_1, \dots, \tau_{[r/2]}$.
- 2: Два варианта критерия остановки следующие. Перебрать элементы множества $i = 1, \dots, [r/2]$.
 1. либо до тех пор, пока $i < m$;
 2. или пока $\tau_i < t_0$.

Обозначим i_0 момент остановки.

- 3: Множество J состоит из индексов $j, j+1$ сингулярных векторов U_j и U_{j+1} , участвующих в вычислении значений $\tau_1, \dots, \tau_{i_0-1}$.
-

Метод может быть применен к правым сингулярным векторам так же, как и для частотного метода. Вычисляем значения $\tilde{\tau}(V_j^{(p)}, V_{j+1}^{(p)})$, $p = 1, \dots, s$, для последовательной пары правых сингулярных векторов $V_j = (V_j^{(1)}, \dots, V_j^{(s)})$ и $V_{j+1} = (V_{j+1}^{(1)}, \dots, V_{j+1}^{(s)})$, $j = 1, \dots, r-1$, форма которых определяется формулой (4.6). Тогда для каждого j берем минимальное значение $\tilde{\tau}^{(\min)}(V_j, V_{j+1}) = \min_p \{ \tilde{\tau}(V_j^{(p)}, V_{j+1}^{(p)}) \}, p = 1, \dots, s \}$ и к уже полученным значениям $\tilde{\tau}^{(\min)}(V_j, V_{j+1})$ применяем стандартный алгоритм, как в одномерном случае. Полученный алгоритм 4.10 показан ниже.

АЛГОРИТМ 4.10: MSSA. Метод идентификации колебательной компоненты по регулярности углов: по правым сингулярным векторам

Input: Следующие данные и параметры являются входными.

1. **Данные:** количество компонентов r ; правые сингулярные векторы $\{V_j\}_{j=1}^r$.
2. **Параметры:** число э.-м. гармоник $m \leq r/2$; или порог $t_0 \geq 0$, в зависимости от критерия остановки.

Output: Группа индексов сингулярных векторов J , связанных с колебательной компонентой.

- 1: Для каждой пары векторов $V_j = (V_j^{(1)}, \dots, V_j^{(s)})$ и $V_{j+1} = (V_{j+1}^{(1)}, \dots, V_{j+1}^{(s)})$ вычислить значения $\tilde{\tau}(V_j^{(p)}, V_{j+1}^{(p)})$, используя (2.27), $j = 1, \dots, r-1$, $p = 1, \dots, s$
- 2: Вычислить $\tilde{\tau}^{(\min)}(V_j, V_{j+1}) = \min_p \{ \tilde{\tau}(V_j^{(p)}, V_{j+1}^{(p)}) \}, p = 1, \dots, s \}$.
- 3: Для каждого $j = 2, \dots, r-1$, если $\tilde{\tau}^{(\min)}(V_j, V_{j+1}) < \tilde{\tau}^{(\min)}(V_{j-1}, V_j)$, то мы отбрасываем $\tilde{\tau}^{(\min)}(V_{j-1}, V_j)$ из рассмотрения; иначе, отбрасываем $\tilde{\tau}^{(\min)}(V_j, V_{j+1})$. Расположим эти значения в порядке возрастания и получим $\tau_1, \dots, \tau_{[r/2]}$.
- 4: Два варианта критерия остановки следующие. Перебрать элементы множества $i = 1, \dots, [r/2]$.
 1. либо до тех пор, пока $i < m$;
 2. или пока $\tau_i < t_0$.

Обозначим i_0 момент остановки.

- 5: Множество J состоит из индексов $j, j+1$ сингулярных векторов U_j и U_{j+1} , участвующих в вычислении значений $\tau_1, \dots, \tau_{i_0-1}$.
-

Глава 5

Многомерный и Shaped SSA

Эта глава посвящена расширениям методов 1D-SSA (главы 2 и 3) и MSSA (глава 4) для анализа объектов размерности 2 и больше. Двумерный случай соответствует обработке цифровых изображений. Объекты с большими размерностями также широко используются. Например, цветное изображение можно рассматривать как систему двумерных изображений и его анализ может быть выполнен с помощью многомерного 2D-SSA, который является расширением MSSA, предназначенного для анализа системы рядов. Третье временное измерение естественно возникает, если изображения изменяются во времени, что является типичным для климатологии [171]. Данные могут быть трехмерными, если измерения проводятся в точках, расположенных в трехмерной области, см. например, раздел 7.5 для количественного анализа данных экспрессии генов. Можно также рассмотреть систему трехмерных данных, изменяющихся во времени. Список расширений можно продолжить; см., например, [75] об алгоритмах SSA для высоких измерений, применяемых к конкретным проблемам в сейсмологии.

Схема методов SSA-семейства из раздела 1.2 может быть естественно применена для многомерных объектов любой размерности, поскольку разница заключается только в операторе вложения. Более того, SSA-анализ объекта любой размерности и формы является частным случаем Shaped SSA, где окно произвольной формы проходит через объект произвольной формы и формирует траекторную матрицу, состоящую из векторизованных запаздывающих окон.

Пакет RSSA реализует так называемый Shaped n D-SSA для анализа объектов произвольных размерностей, прямоугольной и произвольной формы. В частности, двухмерные или трехмерные изображения, которые не обязательно являются прямоугольными и изменяются во времени (то есть имеют дополнительное временное измерение), могут быть проанализированы.

В основу этой главы положена объемная работа [12] и [41]. Мы начинаем главу с описания общего подхода, называемого Shaped SSA, к анализу объектов произвольной формы (Раздел 5.1).

В разделе 5.2 рассматриваются различные методы семейства SSA как частные случаи Shaped SSA, такие как Shaped 1D-SSA, MSSA, 2D-SSA и n D-SSA.

Наконец, в разделе 5.3 мы демонстрируем пример Shaped SSA.

Мы не описываем формально алгоритмы для размерностей больше двух, так как они могут быть получены путем формального расширения алгоритмов из раздела 5.1.

5.1. Shaped 2D-SSA

В общем виде, Shaped SSA может быть использован для анализа объекта, заданного на сетке размером размерностью $k \geq 1$. Случай $k = 1$, который описан в разделе 5.2.1, является, в некоторой степени, вырожденным. Для простоты обозначений, мы рассматриваем случай $k = 2$ как показательный для размерности $k > 1$.

Shaped 2D-SSA (мы будем использовать краткое название *ShSSA*, если размерность понятна по контексту) является обобщением 2D-SSA, которое допускает произвольную форму входного массива и окна. Это значительно расширяет спектр реальных применений SSA, поскольку позволяет анализировать части изображения с различной структурой, исключать области с поврежденными данными, анализировать изображения с пропусками, раскладывать непрямоугольные изображения и т.д. В Shaped SSA не все значения прямоугольного изображения должны быть заданы, а скользящее окно не обязательно прямоугольное. Более того, существует круговая версия Shaped SSA (Circular SSA), когда объект расположен на окружности, цилиндре или торе [41].

5.1.1. Метод

Формы и действия с ними

Назовем *формой* \mathfrak{B} конечное непустое подмножество множества $\{1, \dots, T_x\} \times \{1, \dots, T_y\}$, где $T_x, T_y \in \mathbb{N} \cup \{\infty\}$, \mathbb{N} — множество натуральных чисел. Значения T_x и T_y характеризуют топологию множества, содержащего данную форму. Если $T_x < \infty$ (или $T_y < \infty$), то топология круговая и формы являются циклическими относительно x -координаты (или y -координаты) с периодом T_x (соответственно, T_y).

А \mathfrak{B} -образный массив — это индексированное множество $X = X_{\mathfrak{B}} = (x_{(i,j)})_{(i,j) \in \mathfrak{B}}$. Обозначим пространство \mathfrak{B} -образных массивов как $\mathbb{R}^{\mathfrak{B}}$. Существует изоморфизм $\mathbb{R}^{\mathfrak{B}} \sim \mathbb{R}^B$, где $B = |\mathfrak{B}|$ — мощность множества \mathfrak{B} . Этот изоморфизм не является единственным, так как элементы в форме могут быть упорядочены по-разному. Однако результат алгоритма не зависит от выбранного порядка.

Для удобства мы рассматриваем лексикографический порядок, который фиксирует изоморфизм

$$\mathcal{J}_{\mathfrak{B}} : \mathbb{R}^{\mathfrak{B}} \mapsto \mathbb{R}^B. \quad (5.1)$$

Мы называем $\mathcal{J}_{\mathfrak{B}}$ *векторизацией*, а ее обратную $\mathcal{J}_{\mathfrak{B}}^{-1}$ *формовкой*.

Далее мы вводим операцию сложения в рассматриваемой топологии для двух пар индексов $\ell = (\ell_x, \ell_y)$, $\kappa = (\kappa_x, \kappa_y)$. Для удобства мы опускаем параметры (T_x, T_y) и пишем \oplus вместо $\oplus_{(T_x, T_y)}$:

$$\ell \oplus \kappa = ((\ell_x + \kappa_x - 2) \bmod T_x + 1, (\ell_y + \kappa_y - 2) \bmod T_y + 1),$$

где \bmod обозначает остаток при целочисленном делении. Формально, $a \bmod \infty = a$ для лю-

бого a . Заметим, что -2 и $+1$ в определении \oplus является следствием индексации, которая начинается с 1 (так что $\{1\} \oplus \{1\} = \{1\}$).

Для двух форм \mathfrak{A} и \mathfrak{B} мы модифицируем определение суммы Минковского следующим образом:

$$\mathfrak{A} \oplus \mathfrak{B} = \{\alpha \oplus \beta \mid \alpha \in \mathfrak{A}, \beta \in \mathfrak{B}\}. \quad (5.2)$$

Шаг вложения

Оператор вложения для алгоритма Shaped 2D-SSA определяется следующим образом.

Входные данные и параметры вложения. Входными данными являются характеристики топологии (T_x, T_y) , форма \mathfrak{N} , где $\mathfrak{N} \subset \{1, \dots, T_x\} \times \{1, \dots, T_y\}$, и массив $X \in \mathbb{R}^{\mathfrak{N}}$, заданный на форме \mathfrak{N} . Будем называть такие массивы *фигурными*, переводя таким образом shaped array. Параметром алгоритма является *форма окна* $\mathfrak{L} \subset \mathfrak{N}$. Удобно рассматривать форму окна в виде $\mathfrak{L} = \{\ell_1, \dots, \ell_L\}$, где $L = |\mathfrak{L}|$ и $\ell_i \in \mathbb{N}^2$ упорядочены лексикографически.

Для каждого $\kappa \in \mathfrak{N}$, мы определяем сдвинутый \mathfrak{L} -образный подмассив как $X_{\mathfrak{L} \oplus \{\kappa\}} = (x_\alpha)_{\alpha \in \mathfrak{L} \oplus \{\kappa\}}$. Индекс κ — это положение начала координат для окна. Рассмотрим множество всех возможных положений начала координат для \mathfrak{L} -образных окон:

$$\mathfrak{K} = \{\kappa \in \mathfrak{N} \mid \mathfrak{L} \oplus \{\kappa\} \subset \mathfrak{N}\}.$$

Мы предполагаем, что \mathfrak{K} записывается как $\mathfrak{K} = \{\kappa_1, \dots, \kappa_K\}$, где $K = |\mathfrak{K}| \neq 0$ и κ_j упорядочены лексикографически.

Если формы \mathfrak{N} и \mathfrak{L} являются прямоугольниками, то \mathfrak{K} также является прямоугольной, и мы называем версию Shaped 2D-SSA *прямоугольной*. Заметим, что прямоугольная версия Shaped 2D-SSA является в точности методом 2D-SSA, описанным в разделе 5.2.3.

Оператор вложения. Траекторная матрица \mathbf{X} строится с помощью оператора вложения $\mathcal{T}_{\text{ShSSA}} : \mathbb{R}^{\mathfrak{N}} \rightarrow \mathbb{R}^{L \times K}$.

$$\mathcal{T}(\mathbf{X}) = \mathcal{T}_{\text{ShSSA}}(\mathbf{X}) := \mathbf{X} = [X_1, \dots, X_K], \quad (5.3)$$

где столбцы

$$X_j = (x_{\ell_i \oplus \kappa_j})_{i=1}^L$$

это векторизации подмассивов-форм $X_{\mathfrak{L} \oplus \{\kappa_j\}}$.

Оператор вложения $\mathcal{T}_{\text{ShSSA}}$ является линейным. Обозначим его область значений $\mathcal{M}_{L,K}^{(H)}$ как $\mathcal{H}^{\mathfrak{L} \times \mathfrak{K}} \subset \mathbb{R}^{L \times K}$. Подпространство $\mathcal{H}^{\mathfrak{L} \times \mathfrak{K}}$ состоит из структурированных матриц, которые мы будем называть *квази-ганкелевыми*. (Фактически, они являются обобщением квази-ганкелевых матриц из [174]). Если оператор $\mathcal{T}_{\text{ShSSA}}$ инъективен, то он задает изоморфизм между пространствами $\mathbb{R}^{\mathfrak{N}}$ и $\mathcal{H}^{\mathfrak{L} \times \mathfrak{K}}$.

Замечание 40. *Оператор вложения $\mathcal{T}_{\text{ShSSA}}$ инъективен тогда и только тогда, когда $\mathfrak{L} \oplus \mathfrak{K} = \mathfrak{N}$, то есть, если каждая точка исходного объекта X может быть покрыта окном*

формы \mathfrak{L} .

Если есть непокрытые точки, мы можем удалить их и рассматривать только разложение ограниченного массива $X' = (X)_{\mathfrak{N}'}$, где $\mathfrak{N}' = \mathfrak{L} \oplus \mathfrak{K}$. В дальнейшем мы будем считать, что $\mathfrak{N} = \mathfrak{L} \oplus \mathfrak{K}$, т.е. все точки покрыты и оператор $\mathcal{T}_{\text{ShSSA}}$ инъективен.

Проекция на $H^{\mathfrak{L} \times \mathfrak{K}}$ Проекция $\Pi_{\mathfrak{L}-\text{н} \times \mathfrak{K}}$ на пространство квази-ганделевых матриц порождается оператором вложения $\mathcal{T}_{\text{ShSSA}}$ и может быть выполнена в общем виде, описанном в разделе 1.2.2.

5.1.2. Ранг фигурных массивов

\mathfrak{L} -Rank

Следуя определению временных рядов конечного ранга [1, Глава 5] и двумерных массивов конечного ранга [40] (см. также [39]), давайте обобщим понятие ранга на случай фигурных и круговых форм. Отметим, что в целом теория бесконечных массивов конечного ранга тесно связана с многомерными массивами, удовлетворяющими ЛРС.

Определение 11. \mathfrak{L} -траекторное пространство $S^{\mathfrak{L}}$ *фигурного массива* $S \in \mathbb{R}^{\mathfrak{N}}$ определяется как

$$S^{\mathfrak{L}}(S) = \text{span}\{(S)_{\mathfrak{L} \oplus \{\kappa\}}\}_{\kappa | \mathfrak{L} \oplus \{\kappa\} \subset \mathfrak{N}}.$$

Введенное траекторное пространство соответствует пространству столбцов траекторной матрицы $\mathbf{S} = \mathcal{T}_{\text{ShSSA}}(S)$. Ввиду изоморфизма $\mathcal{J}_{\mathfrak{L}}$, $\{P_1, \dots, P_r\}$ является базисом $S^{\mathfrak{L}}(S)$ тогда и только тогда, когда $\{P_1, \dots, P_r\}$, где $P_k = \mathcal{J}_{\mathfrak{L}}(P_k)$, является базисом пространства столбцов \mathbf{S} . Пусть $\{Q_1, \dots, Q_r\}$ — базис пространства строк \mathbf{S} , то есть траекторного пространства строк. Тогда $\{Q_1, \dots, Q_r\}$, где $Q_k = \mathcal{J}_{\mathfrak{K}}^{-1}(Q_k)$, образует базис пространства, которое можно обозначить как $S^{\mathfrak{K}}(S)$. Мы будем использовать термины *траекторные пространства столбцов/строк* и *фигурные траекторные пространства столбцов/строк*, в зависимости от контекста.

Определение 12. \mathfrak{L} -ранг *фигурного массива* S определяется как размерность его \mathfrak{L} -траекторного подпространства:

$$\text{rank}_{\mathfrak{L}} S = \dim S^{\mathfrak{L}}(S) = \text{rank } S.$$

Бесконечные массивы конечного ранга

Для того чтобы описать общий вид массивов конечного ранга, мы рассмотрим массивы, бесконечные в обе стороны для обоих измерений (т.е. $\mathfrak{N} = \mathbb{Z} \times \mathbb{Z}$), и их траекторные пространства. Рассмотрим плоский случай с $T_x = T_y = \infty$.

Определение 13. Бесконечный (в обоих направлениях) массив S_{∞} называется *фигурным массивом конечного ранга*, если $r = \max_{\mathfrak{L}} \text{rank}_{\mathfrak{L}} S_{\infty}$ (здесь максимум берется над $\mathfrak{L} \subset \{1, \dots, \infty\} \times \{1, \dots, \infty\}$) является конечным. В этом случае мы будем писать $\text{rank } S_{\infty} = r$.

Замечание 41. *Фигурный ранг бесконечного массива равен его прямоугольному рангу (когда берется максимум над прямоугольными фигурами), поскольку последовательность фигур, содержащих возрастающие прямоугольные формы и содержащаяся в прямоугольных формах, может быть легко построена. Поэтому мы будем говорить о массивах конечного ранга, опуская слово “фигура”.*

Замечание 42. *Если S_∞ — бесконечный массив конечного ранга r , $S = (S_\infty)_{\mathfrak{N}}$ — достаточно большой конечный подмассив S_∞ и \mathfrak{L} — достаточно большая форма, то*

$$\text{rank}_{\mathfrak{L}} S = \text{rank}_{\mathfrak{L}} S_\infty = \text{rank } S_\infty = r. \quad (5.4)$$

Пусть $\mathfrak{K} = \{\kappa \mid \mathfrak{L} \oplus \{\kappa\} \subset \mathfrak{N}\}$. Достаточным условием для (5.4) является то, что обе формы \mathfrak{L} и \mathfrak{K} содержат по крайней мере один квадрат $r \times r$. Более формально, достаточно, чтобы существовал двумерный индекс $\alpha = (l, n)$ такой, что $\{1, \dots, r\} \times \{1, \dots, r\} \oplus \{\alpha\} \subset \mathfrak{L}$; то же самое должно быть справедливо для \mathfrak{K} .

Предложение 23. *Пусть $T_x = T_y = \infty$. Тогда бесконечный массив S_∞ конечного ранга имеет вид*

$$(S_\infty)_{m,n} = \sum_{k=1}^s P_k(m, n) \mu_k^m v_k^n, \quad (5.5)$$

где $\mu_k, v_k \in \mathbb{C}$, пары (μ_k, v_k) различны, и $P_k(m, n)$ — комплексные многочлены от m и n . Это представление единственно вплоть до порядка суммирования.

Как было сказано в разделе 1.5, этот важный факт хорошо известен, начиная с [47, §2.20]. Заметим, что ранг массива S_∞ , заданного в (5.5), не определяется только степенями многочленов P_k и его трудно вычислить (см. [40]).

Важным частным случаем является $(S_\infty)_{ln} = \sum_{k=1}^r A_k \mu_k^l v_k^n$; он широко используется в обработке сигналов. В этом случае $\text{rank } S_\infty$ равно количеству различных (μ_k, v_k) -пар.

5.1.3. Алгоритм

АЛГОРИТМ 5.1: Shaped 2D-SSA (планарный): разложение

Input: Изображение X размера $N_x \times N_y$ и маска того же размера состоящая из TRUE и FALSE; окно размером $L_x \times L_y$ и маска того же размера состоящая из TRUE и FALSE; если объекты содержат ‘NA’, это означает, что значения FALSE добавляются к маскам в местах ‘NA’.

Output: Форма \mathfrak{N} , которая была разложена, состоящая из N точек, покрытых окном формы \mathfrak{L} , состоящим из L точек; форма \mathfrak{K} , состоящая из K возможных положений окна. Разложение траекторной матрицы размера $L \times K$: $\mathbf{X} = \mathbf{X}_1 + \dots + \mathbf{X}_d$, где $\mathbf{X}_i = \sqrt{\lambda_i} U_i V_i^T$.

- 1: Построить формы \mathfrak{L} , \mathfrak{K} , \mathfrak{N} и траекторную матрицу $\mathbf{X} = \mathcal{T}_{\text{ShSSA}}(X|_{\mathfrak{N}})$, где $\mathcal{T}_{\text{ShSSA}}$ определяется в (5.3).

2: Вычислить SVD $\mathbf{X} = \mathbf{X}_1 + \dots + \mathbf{X}_d$, $\mathbf{X}_i = \sqrt{\lambda_i} U_i V_i^T$.

Круговая версия 2D-SSA может быть рассмотрена в двух вариантах, цилиндрическом и тороидальном. Прокомментируем случай, когда изображение задано на цилиндре с главной осью, параллельной оси ‘y’. Этот случай соответствует $T_x < \infty$ и $T_y = \infty$. Удобно принять $T_x = N_x$, что означает, что изображение расположено на круговом сегменте цилиндра. Единственное отличие алгоритма кругового 2D-SSA от прямоугольного, см. алгоритм 5.1, заключается в количестве K окон, покрывающих изображение, и, следовательно, в форме траекторной матрицы \mathbf{X} . Например, в цилиндрической топологии окно, которое начинается на одном краю изображения, может продолжаться до противоположного края.

Заметим, что представление U_i или V_i на графиках осуществляется в виде форм $\Psi_i = \mathcal{J}_{\mathfrak{L}}^{-1}(U_i)$ и $\Phi_i = \mathcal{J}_{\mathfrak{K}}^{-1}(V_i)$, которые называются собственными формами и факторными формами соответственно. Здесь \mathcal{J} — оператор формовки, см. (5.1). В цилиндрическом варианте, если окно ограничено, то собственные формы также ограничены, в то время как факторные формы задаются на цилиндре.

Этап восстановления стандартный.

АЛГОРИТМ 5.2: Shaped 2D-SSA: восстановление

Input: Разложение $\mathbf{X} = \mathbf{X}_1 + \dots + \mathbf{X}_d$, где $\mathbf{X}_i = \sigma_i U_i V_i^T$ и $\|U_i\| = \|V_i\| = 1$, группировка $\{1, \dots, d\} = \bigsqcup_{j=1}^m I_j$, \mathfrak{N} .

Output: Разложение объекта на идентифицируемые компоненты: $\mathbf{X}|_{\mathfrak{N}} = \mathbf{X}_1 + \dots + \mathbf{X}_m$.

- 1: Построить сгруппированное разложение матрицы $\mathbf{X} = \mathbf{X}_{I_1} + \dots + \mathbf{X}_{I_m}$, где $\mathbf{X}_I = \sum_{i \in I} \mathbf{X}_i$.
 - 2: $\mathbf{X}|_{\mathfrak{N}} = \mathbf{X}_1 + \dots + \mathbf{X}_m$, где $\mathbf{X}_i = \mathcal{T}_{\text{ShSSA}}^{-1} \circ \Pi_{\mathfrak{N}-\text{нын}}(\mathbf{X}_{I_i})$.
-

5.2. Частные случаи Shaped SSA

В этом разделе мы покажем, что все рассмотренные варианты SSA на самом деле являются частными случаями Shaped SSA, для специальным образом подобранных массива и маски.

5.2.1. Shaped 1D-SSA

Basic SSA соответствует формам

$$\mathfrak{N} = \{1, \dots, N\} \times \{1\}, \quad \mathfrak{L} = \{1, \dots, L\} \times \{1\}.$$

Фигурные массивы $\mathbf{X} \in \mathbb{R}^{\mathfrak{N}}$ в данном случае являются временными рядами, а множество $\mathsf{H}^{\mathfrak{L} \times \mathfrak{K}}$ является множеством обычных ганкелевых матриц.

1D-SSA для временных рядов с отсутствующими значениями получается путем удаления индексов отсутствующих элементов из формы \mathfrak{N} . В этом случае Shaped SSA особенно полезна, когда временной ряд содержит пропуски, так как в этом случае рассмотренные выше версии 1D-SSA не применимы напрямую. Рассмотрим это подробнее.

L -Траекторная матрица $\tilde{\mathbf{X}} = \mathcal{T}_{\text{shSSA}}(\mathbf{X})$ построена так, что ее столбцы являются полными L -запаздывающими векторами. Любые неполные запаздывающие векторы, содержащие недостающие значения, не включаются в $\mathcal{T}_{\text{shSSA}}(\mathbf{X})$.

Обозначим набор элементов ряда, которые представлены в траекторной матрице $\tilde{\mathbf{X}}$, как \mathfrak{N} ; то есть, \mathfrak{N} — это множество непропущенных элементов \mathbf{X} , которые покрыты окнами длины L . Оператор $\mathcal{T}_{\text{shSSA}}$ устанавливает однозначное соответствие между ограничением ряда на \mathfrak{N} и множеством траекторных матриц, если местоположение недостающих данных фиксировано.

Разложение SSA выполняется любым методом (за исключением Toeplitz SSA), описанным в разделах 1.8, 2.1–2.3.1, включая вложенные итеративный O-SSA и DerivSSA. Все эти SSA разложения, за исключением Toeplitz SSA, являются подходящими инструментами, так как построение траекторной матрицы и способ SSA-разложения не влияют друг на друга. Таким образом, мы можем естественным образом определить Shaped Basic SSA, Shaped SSA с проекцией и так далее.

Для Toeplitz SSA теплицево разложение выполняется особым образом, который не основан непосредственно на траекторной матрице; таким образом, версия Toeplitz SSA для объектов произвольной формы не имеет особого смысла и поэтому не реализована в текущей версии RSSA.

Разделимость

Определение и условия разделимости для Shaped SSA такие же, как и для базовых модификаций SSA (см., например, раздел 1.8.3). Однако, точность разделимости естественно, хуже, когда в ряде есть пропуски. Более того, существуют экстремальные случаи, когда ранги рядов могут быть искажены пропусками, и условия разделимости не могут быть выполнены.

5.2.2. MSSA

Представим два способа представления метода MSSA, который был представлен в главе 4 как частные случаи Shaped SSA.

MSSA: 2D-упаковка

Рассмотрим многомерный временной ряд, то есть набор $\{\mathbf{X}^{(p)} = (x_j^{(p)})_{j=1}^{N_p}, p = 1, \dots, s\}$ временных рядов s длины N_p , $p = 1, \dots, s$. Мы строим фигурный массив $\mathbf{X}_{\mathfrak{N}}$ такой, что

$$\mathfrak{N} = \{1, \dots, N_1\} \times \{1\} \cup \{1, \dots, N_s\} \times \{s\} \subset \{1, \dots, \max_p N_p\} \times \{1, \dots, s\}.$$

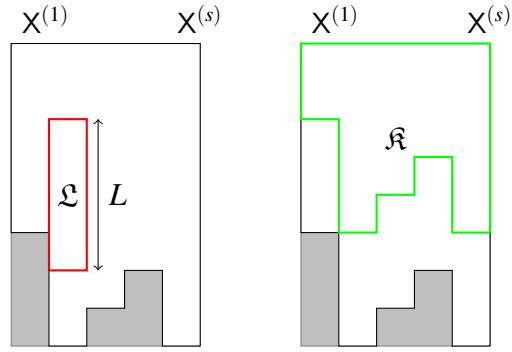


Рис. 5.1: MSSA: 2D упаковка.

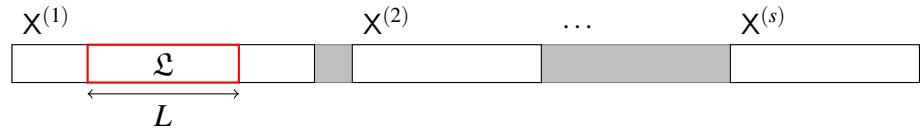


Рис. 5.2: MSSA: 1D упаковка.

Окно принимается равным $\mathfrak{L} = \{1, \dots, L\} \times \{1\}$. Построение этого массива показано на рис. 5.1, также с окном \mathfrak{L} и формой $\hat{\mathfrak{K}}$. Легко проверить, что $\mathbf{X} = \mathcal{T}_{\text{ShSSA}}(\mathbf{X})$ совпадает с $\mathcal{T}_{\text{MSSA}}(\mathbf{X})$, определенным в (4.1). Поэтому, строки \mathbf{X} являются векторизациями $\hat{\mathfrak{K}}$ -образных подмассивов (см. рис. 5.1).

MSSA: одномерная упаковка

Теперь мы рассмотрим альтернативную упаковку MSSA. Из того же набора рядов мы построим $N' \times 1$ (или $1 \times N'$) массив \mathbf{X} , где $N' = N + (s - 1)$ (напомним, что $N = \sum_{p=1}^s N_p$). Массив состоит из временного ряда плюс “разделители” между ними, которые не включены в форму массива. Окно принимается равным $L \times 1$ (или $1 \times L$), в зависимости от выбранного расположения. На рис. 5.2 показан горизонтальный вариант упаковки.

Мозаичные ганкелевы матрицы

Мозаичная ганкелева матрица [175] — это блочная матрица с ганкелевыми блоками. Ее можно считать наиболее общим одномерным (т.е. с одномерным смещением) обобщением ганкелевых матриц.

Пусть L_1, \dots, L_s и K_1, \dots, K_t — целочисленные векторы, а $\mathbf{X}^{(i,j)} \in \mathbb{R}^{L_i+K_j-1}$ — временные ряды. Тогда мозаичная ганкелева матрица строится следующим образом:

$$\begin{pmatrix} H_{L_1, K_1}(\mathbf{X}^{(1,1)}) & \cdots & H_{L_1, K_t}(\mathbf{X}^{(1,t)}) \\ \vdots & \cdots & \vdots \\ H_{L_s, K_1}(\mathbf{X}^{(s,1)}) & \cdots & H_{L_s, K_t}(\mathbf{X}^{(s,t)}) \end{pmatrix}.$$

Обратите внимание, что размеры блоков могут быть разными. Единственным требованием

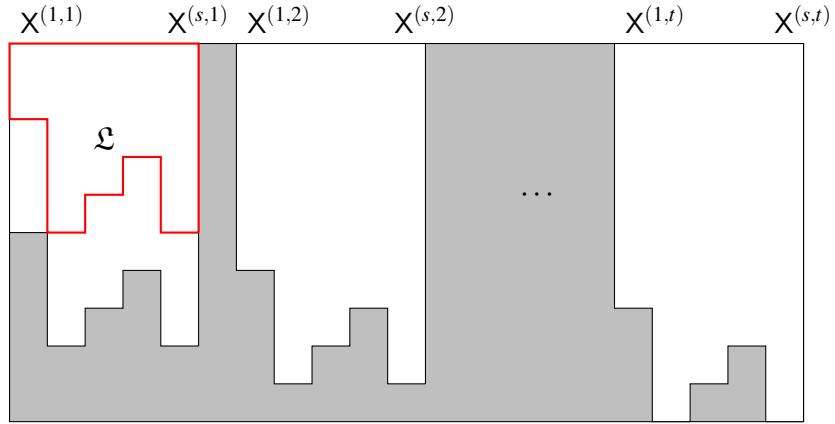


Рис. 5.3: Построение формы для мозаичных ганкелевых матриц.

является то, что они должны совпадать как “мозаика”. Случай мозаичных ганкелевых матриц соответствует нескольким наборам многомерных временных рядов [138].

Легко построить мозаичные ганкелевые матрицы, основываясь на двумерном вложении MSSA. Столбец j -го блока представляет собой транспонированную матрицу $\mathcal{T}_{\text{MSSA}}$ для набора временных рядов $(X^{(1,j)}, \dots, X^{(s,j)})$ и длиной окна $L = K_j$. Таким образом, мозаичная ганкелева матрица может быть построена путем составлением форм (с разделителями) из рис. 5.1 и заменяя \mathfrak{L} на \mathfrak{K} в результате транспонирования. Полученная конструкция фигурного массива показана на Рис. 5.3.

Расширения MSSA на основе подпространства

Ввиду общей структуры всех алгоритмов SSA-семейства (см. раздел 1.2) многие методы, связанные с SSA, могут быть естественным образом распространены с одномерных объектов (т.е. рядов) на другие объекты, и особенно на системы рядов.

В разделе 4.3.1 мы рассмотрели методы MSSA-прогнозирования. Теперь опишем некоторые расширения. Поскольку алгоритмы полностью или почти полностью совпадают с одномерным случаем, мы не будем их описывать. В частности, все методы и алгоритмы, основанные на использовании подпространств столбцов, абсолютно одинаковы в MSSA и 1D-SSA. Например, оценка параметров на основе столбцового подпространства может быть выполнена с помощью одной и той же функции `parestimate` в пакете RSSA.

Shaped версия MSSA почти такая же, как и у 1D-SSA (Раздел 5.2.1) за исключением следующего отличия. Если NA расположены на концах временного ряда, то соответствующий ряд рассматривается как ряд меньшей длины. Если нет других пропущенных данных, то результирующий ряд меньшей длины не имеет пропущенных значений и раскладывается с помощью обычной версии MSSA. Важно отметить, что если в правом конце ряда есть NA, то прогнозирование начинается с последних не-NA значений.

Формально, прогнозирование может быть применено к объектам с пропусками. Однако, как правило, рекомендуется сначала заполнить пропуски (см. раздел 3.1), а затем прогнози-

ровать ряд.

Итеративное заполнение пропусков и низкоранговая аппроксимация итерациями Кэдзуу (раздел 1.9.1) реализованы в общем виде и поэтому могут быть применены к системам временных рядов с пропусками точно так же, как и в одномерном случае.

Заполнение пропусков на основе подпространства имеет больше специфики, однако основано на тех же идеях, что и для одномерного случая.

5.2.3. 2D-SSA

В этом разделе мы рассматриваем расширение алгоритма 1D-SSA для разложения двумерных данных. Это расширение носит название *2D singular spectrum analysis* (или сокращенно *2D-SSA*). Для 2D-SSA объект данных \mathbf{X} представляет собой *двумерный массив данных* размером $N_x \times N_y$ (или просто $N_x \times N_y$ вещественная матрица), представленный в виде $\mathbf{X} = \mathbf{X}_{N_x, N_y} = (x_{ij})_{i,j=1}^{N_x, N_y}$. Типичным примером 2D-массива является цифровое двумерное монохромное изображение.

Алгоритм 2D-SSA соответствует Shaped 2D-SSA с

$$\begin{aligned}\mathfrak{N} &= \{1, \dots, N_x\} \times \{1, \dots, N_y\}, \\ \mathfrak{L} &= \{1, \dots, L_x\} \times \{1, \dots, L_y\}.\end{aligned}$$

Фигурные массивы $\mathbf{X} \in \mathbb{R}^{\mathfrak{N}}$ здесь — это прямоугольные изображения, окна \mathfrak{L} — также прямоугольные, а множество $\mathsf{H}^{\mathfrak{L} \times \mathfrak{K}}$ — это множество *Hankel-block-Hankel* матриц.

2D-SSA был предложен как расширение 1D-SSA в [66], и получил дальнейшее развитие в [40, 176]. Однако, связанные разложения были разработаны независимо в текстурном анализе [177, 178], обработке сейсмических данных [179, 75], и методах оценки параметров для сумм двумерных комплексных экспонент, см. например, [180].

До недавнего времени основным недостатком методов, основанных на разложении 2D-SSA, была их вычислительная сложность. Пакет RSSA содержит эффективную реализацию как общего случая Shaped SSA, так и частного случая 2D-SSA разложения и восстановления, которая в значительной степени преодолевает этот недостаток.

Метод

Для матрицы $\mathbf{A} \in \mathbb{R}^{M \times N}$ (или $\mathbb{C}^{M \times N}$) обозначим через $\text{vec}(\mathbf{A}) \in \mathbb{R}^{MN}$ (или \mathbb{C}^{MN}) ее постолбцовую векторизацию. Для вектора $\mathbf{A} \in \mathbb{R}^{MN}$ (или \mathbb{C}^{MN}), мы определяем его *M девекторизацию*, которую в общем случае называли формовкой, как матрицу $\text{vec}_M^{-1}(\mathbf{A}) = \mathbf{B} \in \mathbb{R}^{M \times N}$ (или $\mathbb{C}^{M \times N}$) которая удовлетворяет $\text{vec}(\mathbf{B}) = \mathbf{A}$.

Оператор вложения Общая схема алгоритма SSA описана в разделе 1.2. Следовательно, чтобы формально представить 2D-SSA, нам нужно только определить оператор вложения

$$\mathcal{T}_{\text{2D-SSA}}(\mathbf{X}) = \mathbf{X}.$$

Параметрами метода являются двумерные *размеры окна* (L_x, L_y) , с ограничениями $1 \leq L_x \leq N_x$, $1 \leq L_y \leq N_y$ и $1 < L_x L_y < N_x N_y$. Для удобства обозначим также $K_x = N_x - L_x + 1$, $K_y = N_y - L_y + 1$. Как и в общей схеме алгоритмов, мы определяем $L = L_x L_y$ (количество строк \mathbf{X}) и $K = K_x K_y$ (количество столбцов \mathbf{X}).

Рассмотрим все возможные $L_x \times L_y$ подматрицы \mathbf{X} (двумерные скользящие окна). Для $k = 1, \dots, K_x$ и $l = 1, \dots, K_y$ определим $\mathbf{X}_{k,l}^{(L_x, L_y)} = (x_{ij})_{i=k, j=l}^{L_x+k-1, L_y+l-1}$ — подматрица размера $L_x \times L_y$, показана на рис. 5.4. Обратите внимание, что ось x направлена вниз, а ось y направлена вправо; начало координат находится в левом верхнем углу. Мы используем эту ориентацию, поскольку она соответствует стандартной математической индексацией матриц.

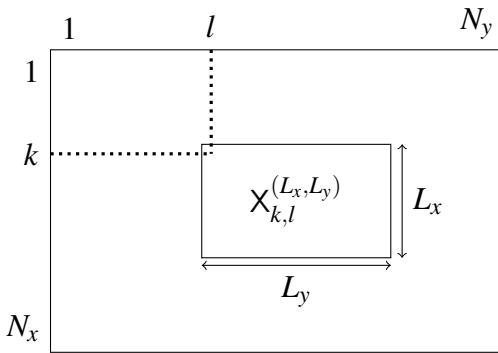


Рис. 5.4: Перемещение двумерных окон.

Тогда траекторная матрица определяется как

$$\mathcal{T}_{\text{2D-SSA}}(\mathbf{X}) = \mathbf{X} = [X_1 : \dots : X_{K_x K_y}], \quad (5.6)$$

где столбцы X_j являются векторизацией подматриц $L_x \times L_y$:

$$X_{k+(l-1)K_x} = \text{vec}\left(\mathbf{X}_{k,l}^{(L_x, L_y)}\right).$$

Структура Hankel-block-Hankel Траекторная матрица (5.6) имеет следующую структуру [40]:

$$\mathbf{X} = \mathcal{T}_{\text{2D-SSA}}(\mathbf{X}) = \begin{pmatrix} \mathbf{H}_1 & \mathbf{H}_2 & \mathbf{H}_3 & \dots & \mathbf{H}_{K_y} \\ \mathbf{H}_2 & \mathbf{H}_3 & \mathbf{H}_4 & \dots & \mathbf{H}_{K_y+1} \\ \mathbf{H}_3 & \mathbf{H}_4 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \mathbf{H}_{L_y} & \mathbf{H}_{L_y+1} & \dots & \dots & \mathbf{H}_{N_y} \end{pmatrix}, \quad (5.7)$$

где каждый \mathbf{H}_j — это $L_x \times K_x$ ганкелева матрица, построенная из $\mathbf{X}_{:,j}$ (j -й столбец двумерного массива \mathbf{X}). Более точно, $\mathbf{H}_j = \mathcal{T}_{\text{1D-SSA}}(\mathbf{X}_{:,j})$, где $\mathcal{T}_{\text{1D-SSA}}$ определено в (1.1). Матрица (5.6) на-

зывается *Hankel-block-Hankel* (сокращенно *HbH*), так как это ганкелевый блок с ганкелевыми блоками. Таким образом, $\mathcal{M}_{L,K}^{(\mathbf{H})}$ в 2D-SSA — это множество матриц *HbH*.

Проекция $\Pi_{\mathbf{H}\times\mathbf{H}}$ на пространство матриц *HbH* может быть выполнена с использованием общей формы, описанной в разделе 1.2.2. В данном конкретном случае проекция состоит из ганкелизации каждого блока \mathbf{H}_j матрицы и затем усреднения блоков в местах \mathbf{H}_j с одинаковым индексом j .

Траекторное пространство Из вида (5.6) мы видим, что траекторное пространство — это линейное пространство, порожденное $L_x \times L_y$ подматрицами матрицы \mathbf{X} . Следовательно, собственные векторы U_i можно также рассматривать как векторизации $L_x \times L_y$ массивов. Их девекторизация обозначается $\Psi_i = \text{vec}_{L_x}^{-1}(U_i)$. Аналогично, строки \mathbf{X} являются векторизацией подматриц размера (K_x, K_y) :

$$\mathbf{X} = [X^1 : \dots : X^{L_x L_y}]^T, \quad X^{k+(l-1)L_x} = \text{vec}(\mathbf{X}_{k,l}^{(K_x, K_y)}), \quad (5.8)$$

где X^j — j -ая строка матрицы \mathbf{X} . Факторные векторы V_i можно также рассматривать как массивы $K_x \times K_y$. Их девекторизация обозначается $\Phi_i = \text{vec}_{K_x}^{-1}(V_i)$.

Комментарии

1. Алгоритм 2D-SSA совпадает с алгоритмом MSSA для временных рядов одинаковой длины, когда $L_x = 1$ или $L_y = 1$.
2. Массивы \mathbf{X} конечного ранга в 2D-SSA (т.е. массивы такие, что $\mathcal{T}_{2D-SSA}(\mathbf{X})$ является матрицей неполного ранга и имеет фиксированный ранг) являются суммами произведений многочленов, экспоненциалов и косинусов, аналогично одномерному случаю. Более подробно это описано в [40].
3. Общая схема SSA, описанная в разделе 1.2, включает в себя разложение траекторной матрицы в сумму матриц единичного ранга. Если это разложение выполняется с помощью SVD, то мы называем метод Basic 2D-SSA или просто 2D-SSA. Другие варианты, такие как вложенный итеративный Oblique 2D-SSA или 2D-SSA с проекцией, также могут быть рассмотрены.

Автоматизация 2D-SSA

Мы рассматриваем поле размером $N_x \times N_y$: $\mathbf{X} = (x_{i,j})_{i,j=1}^{N_x, N_y}$, $x_{i,j} \in \mathbb{R}$.

Для 2D-SSA мы обобщим только низкочастотный метод, описанный для 1D-SSA в алгоритме 2.13 в разделе 2.5.1 [129].

Низкочастотный метод для идентификации паттерна Определим двумерную периодограмму для поля $\mathbb{Y} = (y_{n,m})_{n,m=1}^{M_x, M_y}$ как

$$\Pi_{\mathbb{Y}}^{M_x M_y} \left(\frac{k}{M_x}, \frac{l}{M_y} \right) = M_x M_y \|G_{kl}\|,$$

где $1 \leq k \leq M_x$, $1 \leq l \leq M_y$, G_{kl} — комплексные числа, которые являются коэффициентами двумерного разложения Фурье поля \mathbb{Y} :

$$y_{n,m} = \sum_{k=1}^{M_x} \sum_{l=1}^{M_y} G_{kl} e^{2\pi i (nl/M_x + mk/M_y)},$$

$$G_{kl} = \frac{1}{M_x M_y} \sum_{n=q}^{M_x} \sum_{m=2}^{M_y} y_{n,m} e^{-2\pi i (nl/M_x + mk/M_y)}.$$

Как и в случае 1D-SSA, мы определяем для поля \mathbb{Y} и $-0.5 \leq \omega_1, \omega_2 \leq 0.5$ меру

$$T(\mathbb{Y}; \omega_1; \omega_2) = \sum_{k:0 \leq k/M_x \leq \omega_1} \sum_{l:0 \leq l/M_y \leq \omega_2} I_{\mathbb{Y}}^{M_x M_y}(k/M_x, l/M_y), \quad (5.9)$$

где $I_{\mathbb{Y}}^{M_x M_y}(k/M_x, l/M_y) = \Pi_{\mathbb{Y}}^{M_x M_y} \left(\frac{k}{M_x}, \frac{l}{M_y} \right) / \|\mathbb{Y}\|^2$.

Так как $\|\mathbb{Y}\|^2 = \sum_{k=1}^{\lfloor M_x/2 \rfloor} \sum_{l=1}^{\lfloor M_y/2 \rfloor} \Pi_{\mathbb{Y}}^{M_x M_y} \left(\frac{k}{M_x}, \frac{l}{M_y} \right)$, то меру $T(\mathbb{Y}; \omega_1; \omega_2)$ можно рассматривать как вклад частот, содержащихся в частотном прямоугольнике $\{[0, \omega_1] \times [0, \omega_2]\}$.

Алгоритм 5.3 реализует низкочастотный метод для 2D-SSA.

АЛГОРИТМ 5.3: 2D-SSA. Низкочастотный метод для тренда

Input: Следующие данные и параметры являются входными.

1. **Данные:** индексная группа $I \in \{1, \dots, d\}$; значения $-0.5 \leq \omega_1, \omega_2 \leq 0.5$; массивы \mathbb{Y}_i , $i \in I$, в качестве которых можно брать элементарные восстановленные или векторизованные левые/правые сингулярные векторы.
2. **Параметры:** порог $0 \leq T_0 \leq 1$.

Output: Группа индексов $J \subset I$ компонентов, связанных с трендом.

- 1: Для каждого поля \mathbb{Y}_i , $i \in I$, вычислить значение $T(\mathbb{Y}_i; \omega_1; \omega_2)$, используя (5.9).
- 2: Множество J — это группа индексов $i \in I$ таких, что $T(\mathbb{Y}_i; \omega_1; \omega_2) \geq T_0$.

5.2.4. M-2D-SSA

Предположим, что у нас есть s массивов $\mathbb{X}^{(1)}, \dots, \mathbb{X}^{(s)} \in \mathbb{R}^{N_x \times N_y}$ и мы хотели бы рассмотреть вариант 2D-SSA, где траекторная матрица складывается из s траекторных матриц (как

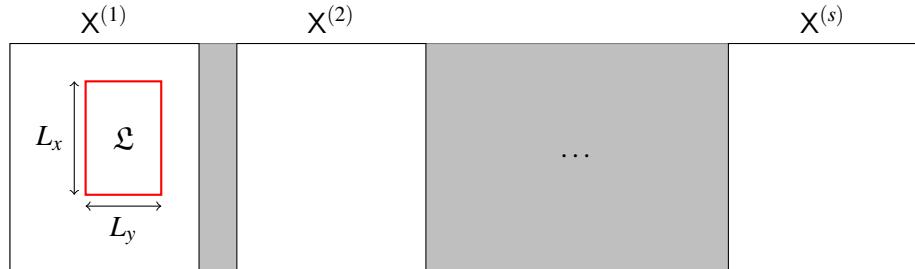


Рис. 5.5: Форма конструкции для M-2D-SSA.

в MSSA):

$$\mathcal{T}_{\text{M-2D-SSA}}(X^{(1)}, \dots, X^{(s)}) = [\mathcal{T}_{\text{2D-SSA}}(X^{(1)}), \dots, \mathcal{T}_{\text{2D-SSA}}(X^{(s)})]. \quad (5.10)$$

В этом случае 2D-SSA-подобное разложение будет иметь общий базис из собственных векторов, как в MSSA. Траекторные матрицы вида (5.10) используются в 2D-SSA для сравнения изображений [176]. Эти матрицы также используются в недавних методах параллельной магнитно-резонансной томографии [181].

Упаковка для M-2D-SSA может быть построена аналогично случаю с мозаичными ганкелевыми матрицами. Массив X' размером $N_x \times (sN_y + s - 1)$ строится из массивов с однодimensionalными разделителями. Полученный массив показан на рис. 5.5. В целом, подобная конструкция может работать с массивами различных размеров, форм и фигурных окон. Также отметим, что в расширенном массиве исходные массивы (как для рис. 5.3 и 5.5) могут быть расположены произвольно (например, в виде таблицы). Единственным требованием является то, что они должны быть разделены.

5.2.5. Комментарии к nD расширениям

В начале этой главы мы упомянули, что случай nD очень похож на случай 2D. Более того, многие подходы, представленные в главах 2 и 3 для случая 1D, могут быть применены и в случае nD. Давайте опишем общие черты и особенности случая nD с $n > 2$.

- nD-Объекты могут быть непрямоугольной формы; также могут использоваться непрямоугольные окна. Т.е., реализуется Shaped nD-SSA. Более того, все другие версии SSA можно рассматривать как частные случаи Shaped nD-SSA.
- Метод nD-ESPRIT оценки частоты и коэффициента затухания, который является прямым продолжением 2D-ESPRIT и Shaped ESPRIT, реализуется для объектов любой размерности и формы.
- Для случая nD итеративный O-SSA и O-SSA с предварительной фильтрацией, которые описаны в главе 2, а также итеративное заполнение пропусков и итерации Кэдзоу, которые описаны в главе 3, реализуются полностью аналогичным образом.

- Визуальное обследование объектов n D является очевидно сложной задачей. Поэтому визуальный контроль результатов в n -мерном случае затруднен.

5.3. Пример Shaped SSA

Рассмотрим пример работы алгоритма Shaped 2D-SSA для удаления текстуры в изображении непрямоугольной формы. В качестве примера мы используем изображение Марса работы Пьера Тьери, из руководства бесплатного программного обеспечения IRIS. Изображение имеет размер 258×275 , 8-битную градацию серого, значения от 0 до 255. Результаты восстановления для разложения прямоугольного изображения показаны на рис. 5.6.

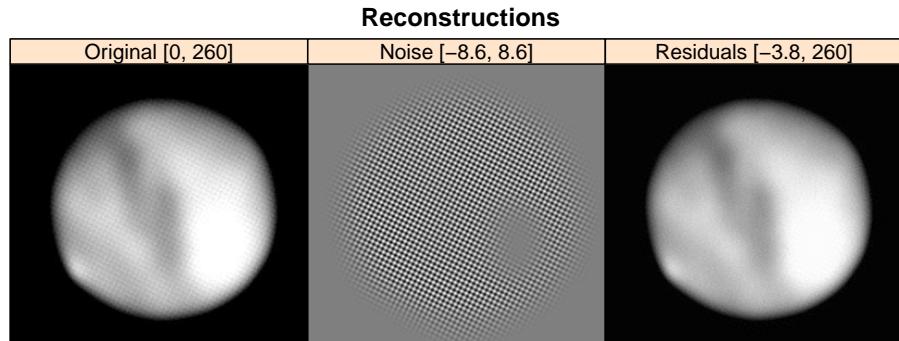


Рис. 5.6: ‘Mars’: Разложение методом 2D-SSA, $(L_x, L_y) = (25, 25)$.

Видно, что изображение Марса имеет круглую форму и есть небольшая область зашивания. Поэтому возникает эффект размывания резких границ. Давайте применим к изображению Shaped 2D SSA. Для этого нам необходимо задать маски.

Форма массива может быть задана двумя различными способами:

- задавая значения `NA` во входном массиве (эти элементы исключаются), или,
- задавая логическую маску в виде логического массива $N_x \times N_y$ (индикатор формы \mathfrak{N}).

Если присутствуют оба способа задания формы, то рассматривается их пересечение.

На рис. 5.7 показаны оба типа масок и комбинированная маска. Результаты восстанов-

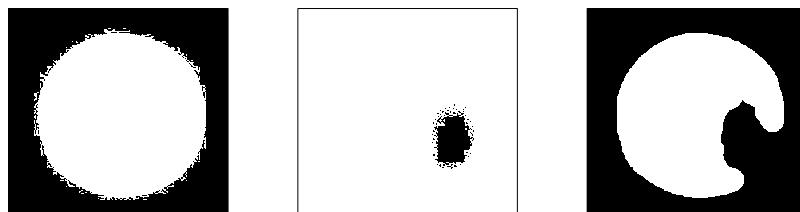


Рис. 5.7: Задание масок для Марса. Слева: маска, заданная с помощью `NA`, в центре: логическая маска, справа: результирующая маска. Белые квадраты — `TRUE`, черные квадраты — `FALSE`

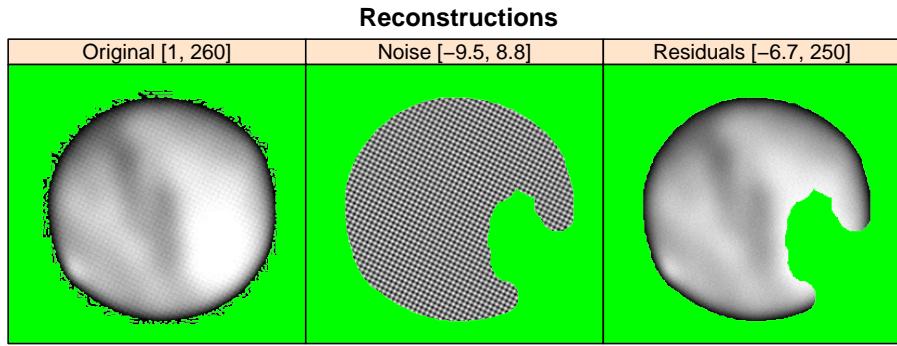


Рис. 5.8: Марс: Разложение методом Shaped SSA, круглое окно радиуса 15.

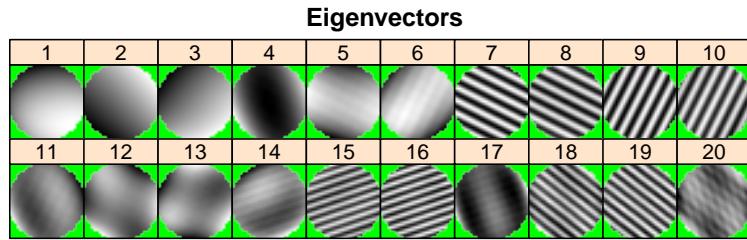


Рис. 5.9: Марс: Собственные массивы, Shaped SSA.

ления вместе с исходным изображением показаны на рис. 5.8. Использовалось круглое окно радиуса 15. На рис. 5.8 видно, что элементы разложения находятся только внутри результирующей маски, однако оригинальный массив нарисован для всех доступных элементов (за исключением значений NA). Группировка для этого разложения была сделана на основе следующей информации:

- собственные массивы (см. рис. 5.9), и,
- матрица \mathbf{w} корреляций (см. рис. 5.10).

Качество выделения текстуры и, следовательно, восстановления изображения с помощью Shaped SSA (рис. 5.8) значительно лучше, чем у 2D-SSA; улучшение точности восстановления объясняется краевым эффектом, который вызван резким падением интенсивности вблизи границы Марса. На рис. 5.11 мы сравниваем увеличенные восстановленные изображения для 2D-SSA и Shaped 2D-SSA. На левом рисунке зеленая тень показана для фоновой области, чтобы обозначить границу Марса. На правом рисунке светло-зеленый цвет соответствует NA .

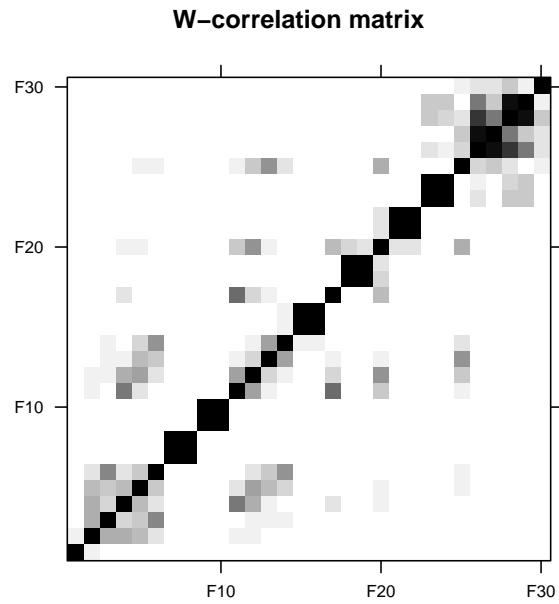


Рис. 5.10: Марс: **w** корреляции, Shaped SSA.

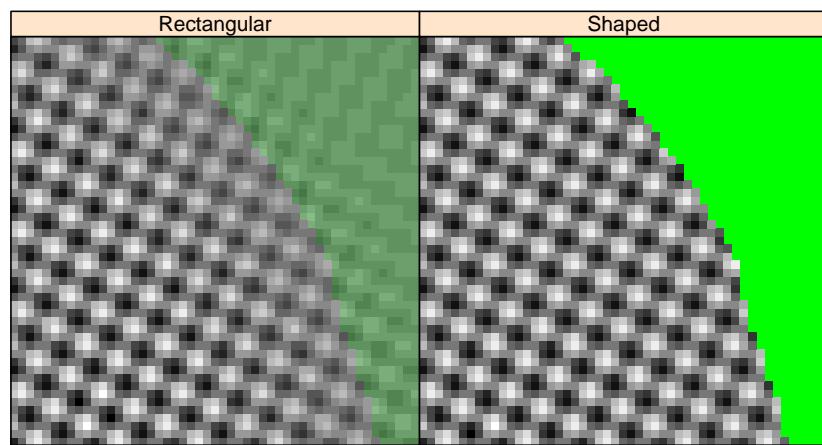


Рис. 5.11: Марс: Сравнение восстановлений текстуры с помощью 2D-SSA и Shaped SSA.

Глава 6

Пакет RSSA

Существует множество реализаций метода SSA. Они различаются по потенциальным областям применения, реализуемым методам, интерактивной и неинтерактивной форме, свободному или коммерческому использованию, операционной системой (Windows, Unix, Mac), уровнем надежности и поддержки. Наиболее известными поддерживающими программными пакетами, реализующими SSA, являются следующие:

1. <http://gistatgroup.com>:
интерактивное программное обеспечение общего назначения ‘Caterpillar’-SSA (Windows), соответствующее методологии, описанной в [1, 29];
2. <http://www.atmos.ucla.edu/tcd/ssa>:
ориентированный в основном на климатические приложения SSA-MTM Toolkit для спектрального анализа [168] (Unix) и его коммерческое расширение kSpectra Toolkit (Mac), интерактивный;
3. Коммерческое статистическое программное обеспечение, SAS, имеет эконометрическое расширение под названием SAS/ETS®, которое включает SSA в его довольно базовой форме; эта версия SSA основана на методологии [1].
4. <http://cran.r-project.org/web/packages/Rssa>:
R-пакет RSSA [70], реализация основных процедур SSA для основных платформ, широко развитая.

Мы рассматриваем пакет RSSA как эффективную реализацию большого количества основных SSA алгоритмов. Этот пакет также содержит множество визуальных инструментов, которые полезны для правильного выбора параметров SSA и изучения результатов.

RSSA — единственный пакет SSA, доступный в CRAN, и мы считаем, что это самая быстрая реализация SSA. Другой важной особенностью пакета является его очень тесная связь с методологией SSA, подробно описанной в [1, 29, 30]. В результате этого, использование пакета имеет хорошую теоретическую и методологическую поддержку.

Вклад автора в создание пакета заключается в разработке общей структуры, создании интерфейсов функций, написании справки к пакету, разработке алгоритмов методов для их реализации вместе с описанием их в [8, 30, 12], посвященным RSSA.

Инструменты для визуального контроля и выбора параметров Для SSA необходимы инструменты, помогающие выбрать параметры методов и контролировать результаты. В значительной степени SSA является исследовательским методом, поэтому визуальные инструменты жизненно необходимы, и они широко используются в RSSA. Например, чтобы

помочь выбрать группы в (1.4), RSSA позволяет строить графики мер разделимости компонентов ряда в полученных разложениях.

Инструменты для контроля точности делятся на две группы. Во-первых, можно проверить стабильность результатов по отношению к изменению параметров. Во-вторых, бутстреп процедура может быть использована, когда модель (не обязательно параметрическая) либо ряда, либо другого объекта строится на основе восстановления сигнала, а затем точность этой модели оценивается путем имитации по оцененной модели. Таким образом, RSSA позволяет пользователю пользоваться большим разнообразием графических инструментов и процедур бутстрата.

6.1. Краткое введение в RSSA

Давайте перечислим общие принципы реализации SSA в RSSA.

- Интерфейс функций для различных типов объектов унифицирован.
- Используется эффективная и быстрая реализация SVD (на C), основанная на едином математическом подходе.
- Многие подходы, впервые разработанные для 1D-SSA, реализованы для многомерных объектов.

Пакет RSSA реализует множество методов и инструментов, разработанных в рамках SSA. Основной функцией является `ssa`, которая инициализирует объект `ssa` и в режиме по умолчанию выполняет разложение различными методами. Вместе с `reconstruct`, они реализуют метод SSA. Вложенные версии разложений, `iossa` и `fossa`, служат для уточненного разложения.

`ssa`-Объект `s` содержит, помимо прочего, элементы разложения (1.2), которые могут быть доступны как `s$sigma`, `s$U` и `s$V`. Особенности разложений отличаются для разных версий SSA (см. раздел 1.3.1). Для Basic SSA `s$sigma` называются сингулярными значениями; квадраты `s$sigma` называются собственными значениями; `s$U` называются собственными векторами. (Мы сохраняем эти названия и для других версий SSA.) Относительные вклады компонентов в разложение можно получить как `contributions(s)`; см. Раздел 1.3.1, где приведены и объяснены формулы для их вычисления.

Разнообразные функции `plot` помогают визуализировать результаты и дополнительную информацию. Функциональность методов, связанных с SSA, дополняется функциями `forecast`, `parestimate` и некоторыми другими.

Все основные версии SSA реализованы в RSSA, но не все дальнейшие действия, такие как прогнозирование и заполнение пропусков, соответствуют всем реализованным версиям SSA. Пользователь может проверить `ssa-object`, который возвращается основной функцией `ssa`,

на согласованность с помощью функции `ssa.capabilities`. Эта функция возвращает список возможностей с информацией TRUE или FALSE соответственно.

Общая схема исследования с помощью RSSA выглядит следующим образом:

1. выполнить разложение по `ssa`;
2. визуализировать результат с помощью `plot`;
3. если необходимо, уточнить разложение по `iossa` или `fossa`;
4. снова визуализировать результат с помощью `plot`;
5. провести группировку на основе полученной визуальной и числовой информации; в частности, выбрать группу компонентов сигнала;
6. затем выполнить одно из следующих действий: восстановление компонентов ряда по `reconstruct`, прогнозирование по `forecast`, оценка параметров по `parestimate`;
7. визуализировать результат с помощью `plot`.

6.2. Эффективность реализации

6.2.1. Эффективность R-пакета RSSA

Пользователю не нужно знать особенности внутренней реализации функций из пакета RSSA. Однако понимание общих принципов реализации может помочь использовать пакет более эффективно.

Быстрая реализация методов, связанных с SSA, которая была предложена в [113], расширена в [12] и используется в пакете RSSA [70], опирается на следующие техники (более подробное обсуждение см. в [41]).

1. Используется усеченное SVD, вычисленное методами Ланцоша [173, Гл. 9]. В большинстве приложений SSA только несколько ведущих компонентов SVD соответствуют сигналу и поэтому используются на этапе группировки алгоритма SSA. Таким образом, для SSA обычно требуется усеченное SVD, а не полное.
2. Методы Ланцоша не используют явное представление разложенной матрицы \mathbf{A} . Им нужны только результаты умножения \mathbf{A} и \mathbf{A}^T на некоторые векторы. Ввиду особой ганкелевой структуры \mathbf{A} в алгоритмах SSA, умножение на вектор может быть реализовано очень эффективно с помощью быстрого преобразования Фурье (FFT). Алгоритмы быстрого SVD реализованы в R-пакете SVD [182] таким образом, что их входом является функция вектора, которая дает результат быстрого умножения вектора на траекторную матрицу. Таким образом, использование SVD в RSSA позволяет быструю и экономичную реализацию алгоритмов SSA.

3. На шаге восстановления, ганкелизация или квази-ганкелизация матрицы ранга 1, хранящейся как $\sigma U V^T$, может быть записана с помощью оператора свертки и поэтому может быть эффективно реализована; это также сделано в RSSA.

По сравнению с прямой реализацией, общая сложность вычислений снижается с $O(N^3)$ до $O(kN \log N + k^2N)$ а расход памяти уменьшается с $O(N^2)$ до $O(N)$, где N — число элементов в объекте, а k — число рассматриваемых собственных троек, см. обоснование в [113] и в [12]. Это делает вычисления выполнимыми для больших наборов данных и больших размеров окна. Например, случай с изображением размером 299×299 и размером окна 100×100 может быть обработан довольно легко, тогда как обычные алгоритмы (например, полное SVD [173]) неприменимы, поскольку матрица, которую нужно разложить, имеет размер $10^4 \times 4 \cdot 10^4$. Использование окон большего размера часто полезно, так как, например, можно значительно улучшить отделимость сигнала от шума (в модели “сигнал+шум”).

6.2.2. Пример вычислений в RSSA

Давайте продемонстрируем, насколько быстрыми являются вычисления в RSSA. Для временного ряда длиной $N = 1000000$ и длины окна $L = 500000$, восстановления синусоидального сигнала по двум ведущим компонентам выполняется за несколько секунд:

```
> library("Rssa")
> N <- 1000000
> signal <- sin((1:N) * 2 * pi / 10)
> ts <- signal + 10*rnorm(1:N)
> system.time(s <- ssa(ts, L = N/2, svd.method = "auto", neig = 2))
user      system      elapsed
1.19        0.16       1.34
> system.time(rec <- reconstruct(s, groups = list(sig = 1:2)))
user      system      elapsed
0.55        0.13       0.67
> max(abs(signal - rec$sig))
[1] 0.0515102
```

6.3. Унифицированный подход для реализации схемы SSA

6.3.1. Основной анализ

Продемонстрируем, что разложение и восстановление выполняются аналогичным образом. Начнем с описания схемы применения Basic SSA для восстановления компонента из одномерного временного ряда. В данном примере тренд восстанавливается с помощью ET1,4,7. Код для временных рядов с пропусками такой же.

```

> F <- co2;
> # F[100:200] <- NA #the code will be the same if to add missing values
> # Perform Basic or Shaped SSA
> s1 <- ssa(F, L = 72)
> # Plots for choosing the components for reconstruction
> plot(s1, type = "vectors", idx = 1:12)
> plot(s1, type = "series", groups = 1:6, layout = c(2, 3))
> plot(wcor(s1, groups = 1:20), scales = list(at = seq(1,20,2)))
> # Reconstruction of the trend
> r1 <- reconstruct(s1, groups = list(c(1, 4, 7)))
> # Plot the extracted trend
> plot(r1, add.residuals = FALSE,
+       plot.method = "xyplot", superpose = TRUE)

```

В многомерных расширениях SSA разложение и восстановление выполняются аналогичным образом. В следующем коде применения MSSA, для краткости, мы опускаем код для идентификации тренда и сезонных компонентов.

```

> wineFortDry <- wine[, c("Fortified", "Drywhite")]
> # Perform Multivariate SSA
> s.wineFortDry <- ssa(wineFortDry, L = 84, kind = "mssa")
> r.wineFortDry <- reconstruct(s.wineFortDry,
+                                groups = list(Trend = c(1, 6),
+                                              Seasonality = c(2:5, 7:12)))
> # Plot the extracted trend
> plot(r.wineFortDry, add.residuals = FALSE,
+       plot.method = "xyplot",
+       superpose = TRUE, auto.key = list(columns = 3))

```

Ниже код Shaped 2D-SSA применяется к прямоугольному изображению, где форма создается битовой маской. Поскольку форма изображения Марса близка к кругу, форма окна также является кругом.

```

> data("Mars", package = "Rssa")
> mask.Mars.0 <- (Mars != 0) #removes black
> mask.Mars.1 <- (Mars != 255)
> Mars[!mask.Mars.0] <- NA
> # Perform Shaped 2D SSA
> s.Mars.shaped <- ssa(Mars, kind = "2d-ssa",
mask = mask.Mars.1, wmask = circle(15))
> # Plots for choosing the components for reconstruction
> plot(s.Mars.shaped, type = "vectors", idx = 1:30,
+       fill.color = "yellow", cuts = 255, layout = c(10, 3),

```

```

+     plot.contrib = FALSE)
> plot(wcor(s.Mars.shaped, groups = 1:30),
+       scales = list(at = c(10, 20, 30)))
> plot(s.Mars.shaped)
> r.Mars.shaped.groups <- list(Noise = c(7, 8, 9, 10))
> #reconstruct and plot the pattern
> r.Mars.shaped <- reconstruct(s.Mars.shaped,
+                               groups = r.Mars.shaped.groups)
> plot(r.Mars.shaped, cuts = 255, layout = c(3, 1),
+       fill.color = "yellow")

```

6.3.2. Различные модификации шага разложения

Подход к применению различных модификаций шага разложения также унифицирован.

Модификации, связанные с различными предположениями о структуре временного ряда, реализуются в функции `ssa`, которая возвращает `ssa`-объект. Например, Toeplitz SSA выбирается параметром `kind = 'toeplitz-ssa'`. Версия ProjSSA выбирается, если дополнительные параметры `column.projector` и `row.projector` заданы.

Модификации для вложенных разложений, таких как Iterative-OSSA или DerivSSA, выполняются после определения компонентов сигнала, чтобы улучшить разделимость. Это может быть сделано с помощью функций `iossa` и `fossa`, применяемых к `ssa`-объекту и компонентам сигнала.

Более того, этот подход работает для многомерных объектов произвольной формы. В частности, вложенные разложения в общем случае выполняются теми же `iossa` и `fossa` функциями.

6.3.3. Методы, основанные подпространстве сигнала

Используя `ssa`-объект и группы собственных троек, можно выполнять такие действия, как заполнение пропусков, оценка параметров и прогнозирование. Эти действия также выполняются унифицированным способом. Однако специфика сигнальных подпространств для различных видов входных объектов накладывает ограничения на применение методов, основанных на подпространстве сигнала.

Глава 7

Приложения к анализу реальных данных

В этой главе мы представляем несколько примеров из реальной жизни, где SSA помогает получить результаты в соответствующей реальной области.

В разделе 7.1 SSA используется для прогнозирования параметров ориентации Земли. Показано, что прогноз, в целом, точнее, чем известные ранее. В разделе 7.2 описано применение SSA для сглаживания функций распределения и оценки плотностей.

Разделы 7.3–7.5 посвящены применению SSA для анализа экспрессии генов. В разделе 7.3 мы показываем, как SSA может помочь в построении параметрической модели данных с последующим анализом динамики на основе оцененных параметров. Разделы 7.4 и 7.5 демонстрируют использование Shaped SSA для подавления шума и выделения характерных признаков в 2D и 3D случаях соответственно.

Кроме результатов, приведенных в разделах 7.3–7.5, отметим следующие. В работе [6] SSA впервые применяется для удаления шума в 2D данных экспрессии генов. В приложении к этой статье подробно описана технология применения 2D-SSA к планарным данным для выделения паттерна. В продолжение этой работы в [26] 2D-SSA применяется для анализа модели шума. В работах [27] и [18] SSA применяется для выделения паттерна из одномерных профилей экспрессии генов; особый акцент ставится на вопросах природы шума, биологической или вызванной препроцессингом данных.

Кроме статей, посвященных анализу экспрессии генов, отметим также работы [43, 44], где SSA применяется для фильтрации в 2D цифровых моделях рельефа.

7.1. Прогнозирование временных рядов ЕОР с помощью сингулярного спектрального анализа

Точное прогнозирование параметров ориентации Земли (ЕОР) важно для повышения точности определения местоположения GPS и навигации спутников Земли. Временные ряды ЕОР включают периодические компоненты сложной структуры. В данном разделе предлагается единый подход к выбору параметров алгоритма прогнозирования SSA для прогнозирования временных рядов ЕОР. Для прогнозирования на 365 дней используются данные временного ряда ЕОР, опубликованные IERS в бюллетене 14 C04. Прогнозы, выполненные с помощью предлагаемых методов, сравниваются с прогнозами, полученными из доступных открытых источников.

Параметры ориентации Земли — это набор параметров, которые описывают различные аспекты вращения Земли. Они изменяются во времени и поэтому могут рассматриваться как временные ряды. Во многих приложениях, таких как геолокация или высокоточная спутниковая навигация, необходимы не только наблюдаемые значения ЕОР, но и их прогнозы на

несколько дней вперед. А именно, интерес представляют пять временных рядов: координаты полюса (x, y), продолжительность дня LOD , и смещения небесного полюса (dX, dY). Координаты (x, y) полюса небесной эфемериды определяют положение небесного полюса на поверхности Земли. Полюс движется медленно, поскольку ось мгновенного вращения Земли не остается неподвижной. Универсальное время ($UT1$) — это время земных часов, которые совершают один оборот примерно за 24 часа. Избыточное время оборота называется длиной дня (LOD). Оно подвержено влиянию ветра и движению мирового океана, поэтому имеет сложную структуру и не поддается точному моделированию. Положение небесного полюса описывается моделью прецессии и нутации IAU2000 (Международного астрономического союза). Точность аппроксимации высока, однако, некоторые нутации все еще не предсказуемы. Наблюдаемые поправки к смоделированным координатам небесного полюса в небесной системе отсчета представляют собой смещения (dX, dY).

Кампания по сравнению прогнозов параметров ориентации Земли (ЕОР РСС, 2005–2008) была организована для сравнения существующих методов прогнозирования за один и тот же период времени для различных длительностей прогнозов: 10, 30 и 500 дней. Результаты кампании [183] показали, что меньшие ошибки прогноза могут быть достигнуты путем объединения различных моделей. Аналогичная инициатива была предпринята Международной службой вращения Земли (IERS) в 2010–2015 годах; к сожалению, сайт проекта и его архив в настоящее время недоступны.

К проблеме прогнозирования временных рядов ЕОР применялись различные методы. Интерполяция методом наименьших квадратов с помощью гармонической модели и авторегрессионного прогнозирования (LS+AR) была применена к временным рядам x, y и LOD в [184, 185]. Сезонная авторегрессионная модель была предложена для прогнозирования dX и dY в [186].

Анализ сингулярного спектра (SSA) был использован в нескольких недавних работах для прогнозирования движения полюса (x, y). Применимость метода к данной проблеме была продемонстрирована в работе [187], где параметры были фиксированы и выбраны вручную. В статье [188] предлагается более гибкий подход с комбинацией SSA и анализа на основе копулы.

Следуя работе [28], мы представляем технику для (почти) полностью автоматизированной процедуры выбора параметров для каждого из пяти прогнозов временного ряда ЕОР, которые выполняются на основе исторических данных с помощью анализа сингулярного спектра. Набор параметров состоит как из параметров SSA, так и из длины временного ряда, используемой для прогнозирования. Используемый подход к выбору параметров широко применяется в машинном обучении; поэтому особенно интересно отметить, что точность прогнозирования оказывается сопоставимой с другими публичными прогнозами, что и продемонстрировано в данной работе.

Общие соображения о выборе параметров SSA

Безусловно, точность прогнозирования зависит от правильного выбора параметров, который обсуждался в разделе 3.4. SSA можно рассматривать как параметрический и непараметрический метод. Если траекторная матрица сигнала S_N имеет неполный ранг, то сигнал называется временным рядом конечного ранга. Временные ряды конечного ранга (при некоторых нежестких ограничениях) могут быть точно спрогнозированы с помощью методов прогнозирования SSA. Например, синусоиды и их суммы имеют конечный ранг. Для SSA-прогнозирования периодических рядов не требуется, чтобы период был известен заранее благодаря непараметрической природе SSA. Выбор $L \sim N/2$ рекомендуется для прогнозирования сигналов конечного порядка в присутствии шума.

Если сигнал представляет собой амплитудно-модулированную синусоиду, то он не обязательно будет иметь строго конечный ранг. Однако такой сигнал может быть извлечен с хорошей точностью путем увеличения числа r выбранных компонентов и/или уменьшения длины окна L .

В интерактивной версии SSA идентификация компонентов сигнала осуществляется путем анализа собственных троек разложения. Однако, если временной ряд достаточно длинный, можно применить некоторую автоматическую процедуру выбора L и r для минимизации ошибок прогнозирования на выбранных исторических данных; например, [30, Фрагменты 3.5.13–3.5.15]. Более того, если временной ряд длинный и имеет меняющуюся структуру, возникает вопрос о выборе оптимальной длины временного ряда, используемой для прогнозирования. Проблема выбора оптимальной длины временного ряда может быть решена автоматически только для очень длинного временного ряда.

7.1.1. Источники данных

В качестве исходных данных мы используем данные бюллетеня IERS 14 C04, публикуемые IERS [189]. Этот источник содержит ежедневные данные для каждого из временных рядов EOP, начиная с 1 января 1962 года для x, y, LOD и с 1 января 1984 года для dX, dY до текущей даты с задержкой примерно в 30 дней. Графики временных рядов показаны на рис. 7.1–7.3.

Бюллетень A IERS содержит прогнозы временных рядов x, y, LOD на 365 дней; эти данные публикуются еженедельно. Для сравнения прогнозов мы также будем использовать ежедневные прогнозы Пулковской обсерватории всех временных рядов EOP на 365 дней [190]. К сожалению, по состоянию на июнь 2019 года онлайн-архив с прогнозами Пулково недоступен. Поэтому в данной работе мы будем использовать резервную копию тех файлов, которыми располагаем на данный момент.

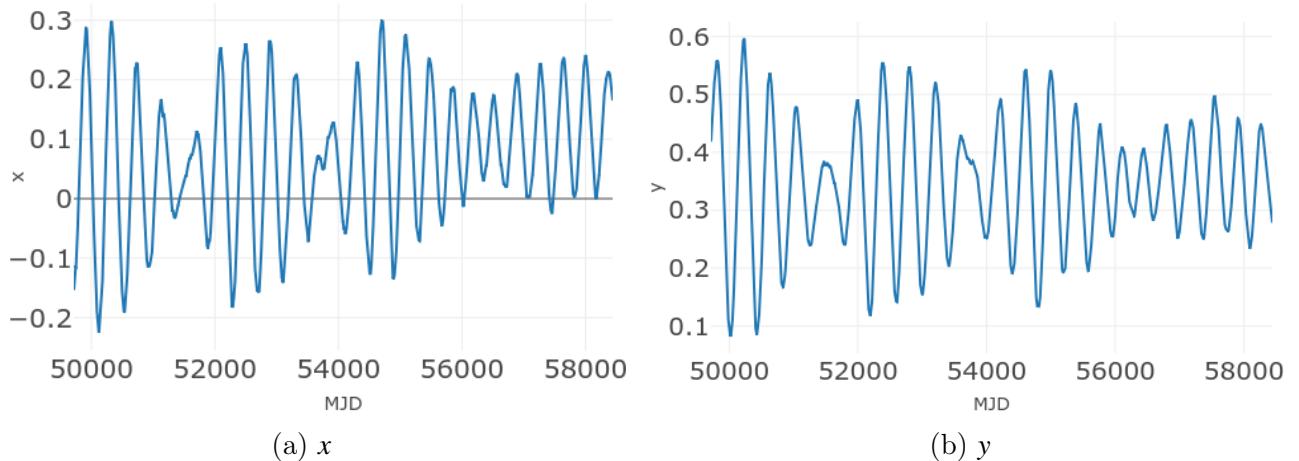


Рис. 7.1: Примеры временных рядов движения небесных полюсов с 1995 года по настоящее время.

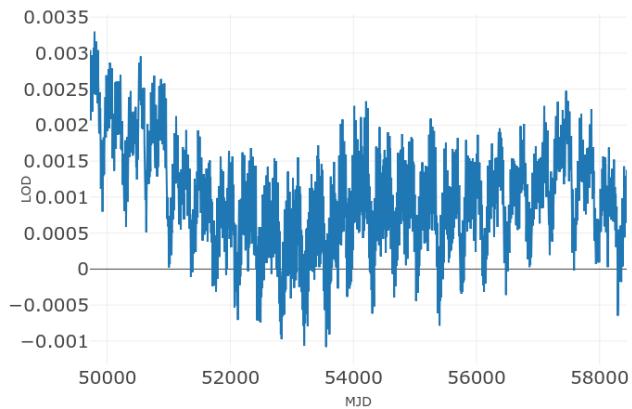


Рис. 7.2: Пример временного ряда *LOD* с 1995 года по настоящее время.

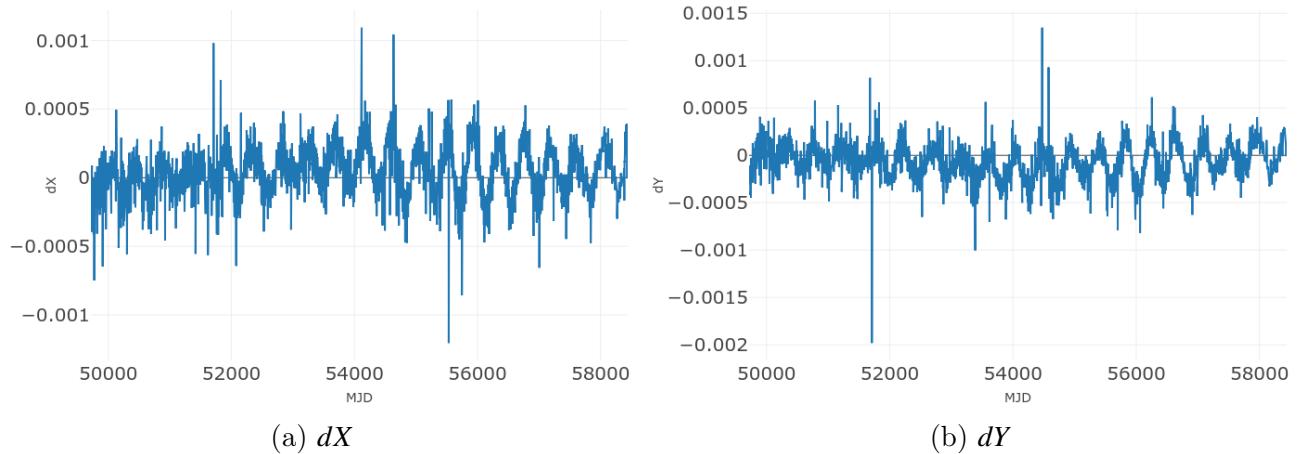


Рис. 7.3: Примеры временных рядов смещений небесных полюсов с 1995 года по настоящее время.

7.1.2. Автоматический выбор параметров

Мы будем выполнять прогнозирование временного ряда EOP x, y, LOD, dX, dY на один год. Для простоты мы считаем, что один год состоит из 365 дней.

Как мы уже говорили, для SSA-прогнозирования сигнала необходимы два параметра: длина окна L и количество ведущих компонент r . Последнее связано со структурой сигнала, которая определяется рангом сигнала. Следующие утверждения связаны: чем сложнее структура сигнала, тем больший ранг используется для его аппроксимации и тем большее число компонент r следует выбирать в SSA-разложении. Мы будем искать оптимальное значение r^* , которое минимизирует среднюю квадратичную ошибку прогноза (MSE) для исторических данных. Рассматриваются границы $r^* \leq 30$ для временных рядов x, y, LOD , и $r^* \leq 5$ для временных рядов dX, dY , так как они имеют более простую структуру. Эти границы были проверены путем ручного SSA-анализа нескольких выборочных данных. Границы для L были выбраны аналогичным образом.

Мы используем поиск оптимального значения L^* по сетке, выполняя до 10 шагов в пределах диапазона. Выбранная сетка для значений L и r показана в таблице 7.1.

Таблица 7.1: Сетка значений параметров L и r .

EOP	значения L	значения r
x, y	300, 500, 700, 900, 1100, 1300, 1500, 1700, 1900, 2100	[1, 30]
LOD	300, 600, 900, 1200, 1500, 1800, 2100, 2400, 2700, 3000	[1, 30]
dX, dY	250, 300, 350, 400, 450, 500	[1, 5]

Для выбора параметров L^*, r^* для прогнозирования SSA мы будем использовать процедуру кросс-валидации временных рядов, как описано в [30, Раздел 3.5.7].

Для проведения кросс-валидации мы должны установить период обучения. В рамках периода обучения мы рассматриваем скользящие интервалы длиной $Q + 1$ лет, где интервал в Q лет ($365 \times Q$ дней) используется для оценки сигнала и один год (365 дней) берется для прогнозирования сигнала и расчета ошибки. Длина периода обучения равна Q лет плюс длина W периода валидации; период валидации состоит из значений временного ряда, которые используются для расчета ошибок прогнозирования, участвующих в процедуре кросс-валидации. Для снижения вычислительных затрат мы рассматриваем M скользящих интервалов длиной $Q + 1$ лет с равным запаздыванием; Таким образом, размер задержки примерно равен $(W - 1)/(M - 1)$. Для прогнозирования SSA, значения L^*, r^* выбираются из сетки, чтобы минимизировать среднеквадратическую ошибку (MSE) прогноза усредняя по всем скользящим интервалам.

В [185] рассматривались значения $Q = 5, 10, 15$ лет. Мы сравниваем $Q = 5, 10, 15, 20$ лет (если длина временного ряда позволяет такой выбор) при длине периода валидации, равной 5 годам. Оптимальные параметры L^*, r^* выбираются автоматически с помощью процедуры кросс-валидации, как описано выше, с $M = 10$.

Использовалась реализация алгоритмов SSA на языке R из пакета RSSA [70]. Средние среднеквадратичные ошибки прогнозов приведены в таблице 7.2. Среди рассмотренных значений значение $Q = 15$ лет приводит к меньшим ошибкам прогноза. Это значение мы будем использовать позже для выполнения прогнозов на тестовый период.

Таблица 7.2: Среднее значение MSE прогнозов EOP на 365 дней для различных Q в интервале 2006–2010 гг.

EOP	5 лет	10 лет	15 лет	20 лет
x	2.6×10^{-3}	9.5×10^{-4}	8.5×10^{-4}	1.1×10^{-3}
y	1.9×10^{-3}	1.2×10^{-3}	9.0×10^{-4}	1.1×10^{-3}
LOD	2.4×10^{-7}	1.9×10^{-7}	1.0×10^{-7}	1.0×10^{-7}
dX	2.01×10^{-8}	1.93×10^{-8}	1.88×10^{-8}	-
dY	1.63×10^{-8}	1.53×10^{-8}	1.42×10^{-8}	-

Таблица 7.3: Среднее значение MSE прогнозов EOP на 365 дней для различных длительностей периода валидации в интервале 2006–2010 гг.

EOP	3 года	5 лет	7 лет	10 лет
x	1.0×10^{-3}	8.5×10^{-4}	8.3×10^{-4}	9.1×10^{-4}
y	9.2×10^{-4}	9.0×10^{-4}	8.8×10^{-4}	8.8×10^{-4}
LOD	1.1×10^{-7}	1.0×10^{-7}	1.0×10^{-7}	9.8×10^{-8}
dX	1.91×10^{-8}	1.88×10^{-8}	1.88×10^{-8}	-
dY	1.42×10^{-8}	1.42×10^{-8}	1.39×10^{-8}	-

Таблица 7.4: Среднее значение MSE еженедельных прогнозов EOP на 365 дней из разных источников в интервале 2011–2015 гг.

EOP	SSA	Pulkovo AM	Bulletin A
x	7.2×10^{-4}	8.6×10^{-4}	7.5×10^{-4}
y	6.1×10^{-4}	7.6×10^{-4}	8.5×10^{-4}
LOD	9.1×10^{-8}	1.0×10^{-7}	-
dX	1.3×10^{-8}	1.1×10^{-8}	-
dY	1.6×10^{-8}	2.2×10^{-8}	-

Нам также нужно выбрать длину периода проверки. Для этого Q фиксируется равным значению, выбранном на предыдущем шаге (15 лет), а затем ошибки прогноза сравниваются с помощью кросс-валидации с $M = 10$ для различных длин периода валидации: 3, 5, 7 и 10 лет. Результаты экспериментов представлены в таблице 7.3. Мы также проверили (не показано), что оптимальные значения r^* и L^* попадают в одни и те же интервалы для разных прогнозов на протяжении рассматриваемого периода обучения. Это подтверждает, что их выбор не является случайным.

7.1.3. Прогнозы на тестовом периоде

Для выполнения прогнозов на 365 дней в тестовом периоде (2011–2015 годы) мы использовали результаты исследований, описанные в разделе 7.1.2. Длительности периодов обуче-

ния и проверки были выбраны с помощью таблиц 7.2 и 7.3, соответственно. Для прогнозов параметры L^*, r^* выбирались автоматически с помощью процедуры перекрестной валидации, описанной в разделе 7.1.2 (примеры прогнозов см. на рис. 7.4). Точность прогноза сравнивалась с точностью прогнозов из двух источников, доступных в открытом доступе: Пулковской обсерватории [190] и Бюллетеня А IERS [191].

Поскольку прогнозы Бюллетеня А публикуются раз в неделю, мы генерируем прогнозы каждую неделю, начиная с дат публикаций для каждого из временных рядов ЕОР. Затем мы рассчитываем среднеквадратичные ошибки MSE для всех прогнозов и берем среднее значение MSE за тестовый период. Результаты показаны в таблице 7.4. В большинстве случаев предложенный метод демонстрирует лучшую среднюю точность, за исключением прогнозов временного ряда dX , опубликованных Пулковской обсерваторией. Однако средние значения MSE прогнозов по SSA и прогнозов Пулковской обсерватории для dX имеют одинаковый порядок.

7.1.4. Выводы

В этой работе мы применили рекуррентный алгоритм прогнозирования SSA для прогнозирования временного ряда ЕОР. Мы сформулировали задачу как задачу машинного обучения и нашли набор параметров обучения, включая гиперпараметры кросс-валидации. Предложенный подход может быть применен к каждому из временных рядов ЕОР: x , y , LOD , dX и dY одинаковым образом.

Оказалось, что SSA хорошо подходит для прогнозирования временных рядов ЕОР. Поскольку вычислительные затраты SSA значительно уменьшились в недавней реализации [70, 30], предлагаемый подход является вычислительно осуществимым для ежедневного прогнозирования временного ряда ЕОР. Численное сравнение в тестовом периоде с 1 января 2011 года по 31 декабря 2015 года показывает, что SSA вместе с предложенным методом автоматизации выбора параметров обеспечивает точность прогноза, сравнимую с теми, которые публикуются международными службами; во многих случаях ошибки предлагаемого метода в среднем меньше.

Полученные прогнозы и соответствующие значения параметров опубликованы в виде веб-приложения и могут быть загружены через <http://eoppredict.ru>. Интерфейс позволяет сравнивать прогнозы из разных источников на исторических данных.

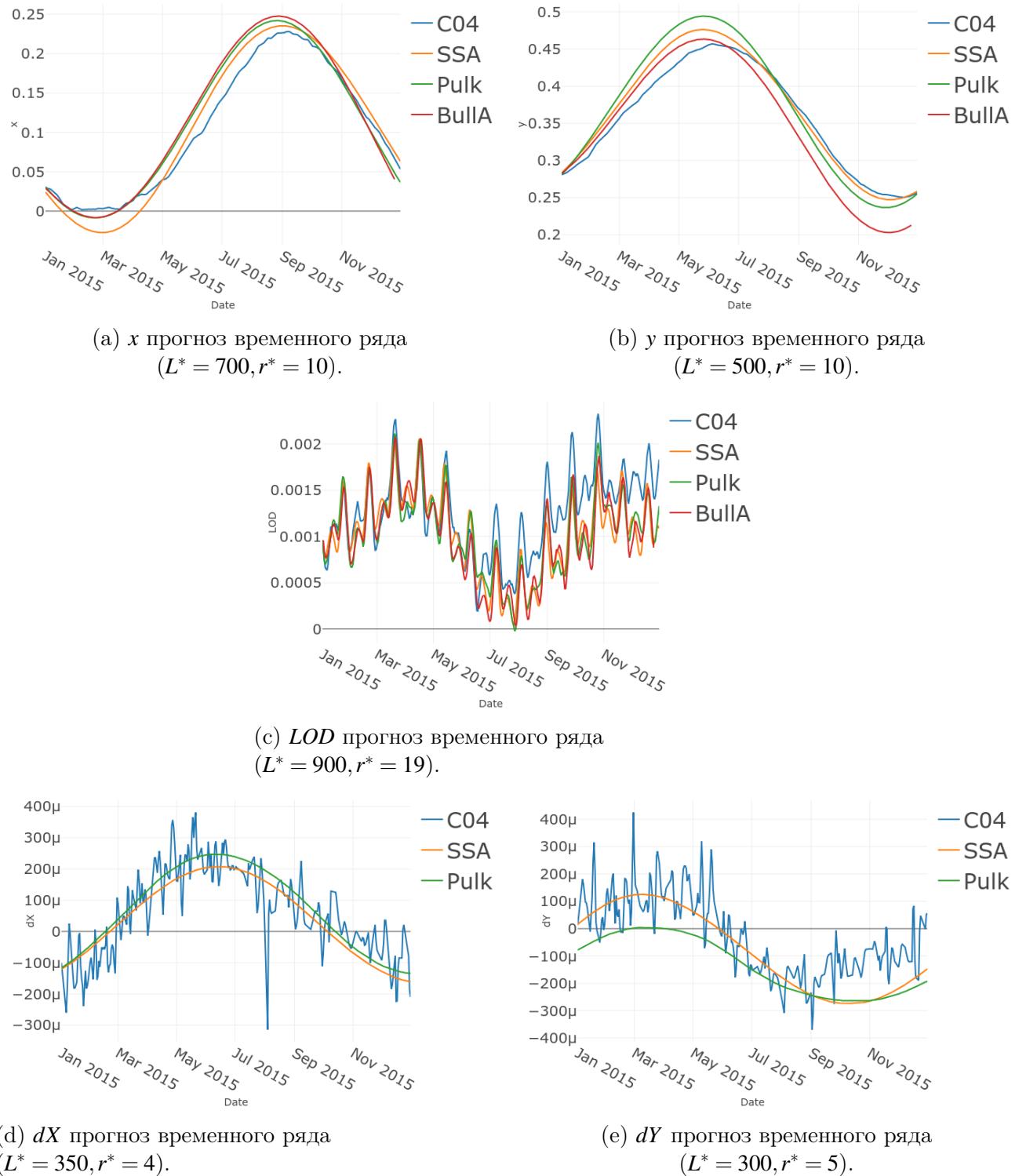


Рис. 7.4: Прогнозы временного ряда ЕОР на 365 дней, начиная с 1 января 2015 года, из разных источников.

7.2. Применение SSA к оценке плотности

Для решения фундаментальной задачи непараметрического оценивания плотности распределения в литературе были предложены различные подходы. Наиболее широко используемым подходом является ядерное сглаживание, подробно рассмотренное в [192, 193].

Цель данного примера — показать, как SSA может быть использован для сглаживания плотности. Изложим новый новый подход к непараметрическому оцениванию функций распределения и плотностей, следя [7]. Основная идея заключается в построении искусственного временного ряда путем оценки эмпирической функции распределения в равноудаленных точках и рассмотрении его как временной ряд.

7.2.1. Новый подход к оценке плотности

Пусть x_1, \dots, x_m — выборка из распределения, имеющего плотность $p(x)$, и пусть $F(x)$ — соответствующая функция распределения. Эмпирическая функция распределения

$$F_m(x) = \frac{1}{m} \sum_{i=1}^m \mathbf{1}_{[x_i, \infty)}(x), \quad x \in \mathbb{R},$$

является несмещенной непараметрической оценкой $F(x)$ с минимальной дисперсией. Однако, она не может быть использована непосредственно для оценки плотности $p(x)$ путем взятия производной, поскольку $F_m(x)$ является ступенчатой функцией; также известно, что оценку часто можно существенно улучшить используя сглаженные, но смещенные оценки $F(x)$. Для сглаживания эмпирической функции распределения, мы предлагаем рассматривать значения $F_m(x)$ на равноотстоящей частой сетке как ряд, а затем применить метод SSA для сглаживания этого ряда.

Обзор подхода

Пусть t_1, \dots, t_N — равноудаленные точки, такие что $t_j - t_{j-1} = \delta$, $j = 2, \dots, N$, $t_1 < \min_i x_i$ и $t_N > \max_i x_i$, где δ — небольшая положительная константа. Определим ряд

$$\mathcal{F}_m = (f_1, \dots, f_N), \quad f_j = F_m(t_j), \quad j = 1, \dots, N,$$

который является неубывающим и служит дискретной версией $F_m(x)$, и ряд

$$\mathcal{F}^* = (f_1^*, \dots, f_N^*), \quad f_j^* = F(t_j), \quad j = 1, \dots, N,$$

который должен быть оценен по выборке.

Теперь выберем некоторое положительное целое число L , удовлетворяющее $1 \leq L \leq (N+1)/2$, которое будет регулировать гладкость и является параметром следующей процедуры, адаптированной из теории SSA, см. [1]. Если рассматривать процедуру сглаживания с точки

зрения фильтрации, L управляет полосой пропускания фильтра. Также параметр L имеет смысл разрешения, то есть, чем меньше L , тем более тонкое и менее устойчивое сглаживание выполняется. Естественной верхней границей для выбора параметра сглаживания L является

$$L_{\max} = \left\lfloor \frac{\max_i x_i - \min_i x_i}{2\delta} \right\rfloor.$$

Заметим, что ряд \mathcal{F}_m может иметь любое количество нулей слева и любое количество единиц справа. Расширение ряда \mathcal{F}_m позволяет нам устраниить граничные эффекты фильтров, построенных далее.

Алгоритм SSA для обработки ряда \mathcal{F}_m теперь работает следующим образом. Как и в базовом алгоритме SSA, строится траекторная матрица и ее сингулярное разложение. Согласно теории SSA, несколько ведущих членов сингулярного разложения представляют интерес для проблемы сглаживания [1, Раздел 1.3.2]. Таким образом, мы определяем матрицу

$$\mathbb{X}^{(r)} = (x_{i,j}^{(r)}) = \sum_{i=1}^r \sqrt{\lambda_i} U_i V_i^T. \quad (7.1)$$

Последним шагом алгоритма SSA является ганкелизация (усреднение по антидиагоналям) $\mathbb{X}^{(r)}$. с последующей операцией, которая противоположна шагу вложения. Учитывая особенную структуру функций распределения, мы определяем

$$\hat{f}_j = \hat{f}_j(L, r) = \begin{cases} 0 & 1 \leq j < L, \\ \frac{1}{L} \sum_{k=1}^L x_{k,j-k+1}^{(r)} & L \leq j \leq K, \\ 1 & K < j \leq N, \end{cases}$$

Таким образом, мы получаем ряд

$$\hat{\mathcal{F}}_m = \hat{\mathcal{F}}_m(L, r) = (\hat{f}_1(L, r), \dots, \hat{f}_N(L, r)),$$

который будем называть оценкой SSA для \mathcal{F}^* . По построению, мы имеем $\hat{\mathcal{F}}_m(L, r) = \mathcal{F}_m$, если $L = r$.

Ряд $\hat{\mathcal{F}}_m$ может быть преобразован в SSA оценку $F(x)$ с помощью линейной интерполяции следующим образом

$$\hat{F}_m(x) = \sum_{j=2}^N \left(\hat{f}_{j-1} + (\hat{f}_j - \hat{f}_{j-1}) \frac{x - t_{j-1}}{t_j - t_{j-1}} \right) \mathbf{1}_{[t_{j-1}, t_j)}(x) + \mathbf{1}_{[t_N, \infty)}(x), \quad (7.2)$$

$x \in \mathbb{R}$. Заметим, что $\hat{F}_m(x)$ на самом деле является линейным сплайнном на $[t_1, t_N]$ с узлами $(t_1, \hat{f}_1), \dots, (t_N, \hat{f}_N)$. Более того, мы можем рассматривать производную $\hat{F}_m'(x)$ слева как оценку

плотности. В частности,

$$\hat{p} = (\hat{p}_1, \dots, \hat{p}_N), \quad \hat{p}_j = (\hat{f}_j - \hat{f}_{j-1})/\delta,$$

$j = 2, \dots, N$, дает оценку плотности $p(x)$ в равноудаленных точках.

Давайте теперь опишем, как выбрать точки t_1, \dots, t_N , используемые для преобразования эмпирической функции распределения в ряд. Для этого сначала заметим, что $V_i = \mathbb{X}^T U_i / \sqrt{\lambda_i}$ и $\mathbb{X}^{(r)} = (U_1 U_1^T + \dots + U_r U_r^T) \mathbb{X}$, и, следовательно, \hat{f}_j допускает представление

$$\hat{f}_j = \sum_{i=1}^L \sum_{l=1}^L (u_{1,i} u_{1,l} + \dots + u_{r,i} u_{r,l}) f_{j+i-l} / L \quad (7.3)$$

для $j = L, \dots, K$, где $(u_{l,1}, \dots, u_{l,L})^T = U_l$. Таким образом, формула (7.3) означает, что процедура SSA создает адаптивный к данным фильтр размером $2L-1$, где r можно интерпретировать как сложность фильтра SSA, а L управляет гладкостью отфильтрованного ряда. Таким образом, мы должны определить точки t_1, \dots, t_N такие, что ряд \mathcal{F}_m содержит $2L$ нулей в начале и $2L$ единиц в конце.

Мы предлагаем выбрать δ так, чтобы удовлетворять неравенству

$$\mathbf{P}(\xi \in [x, x + \delta]) \approx p(x)\delta < 1/m,$$

где плотность $p(x)$ предполагается достаточно гладкой. Тогда

$$\delta \leq \frac{1}{m \max_x p(x)} = \frac{\sigma}{m \max_x p_{st}(x)}, \quad (7.4)$$

где $p_{st}(x)$ — это стандартизированная плотность с нулевым средним и единичной дисперсией, $p_{st}(x) = p((x - a)/\sigma)$. Это неравенство обеспечивает разумное соответствие между размером выборки и шагом интерполяции δ . В примерах данной работы мы рассматриваем $\delta = 0.01\sigma$, что приближенно удовлетворяет неравенству (7.4).

Обобщая вышеприведенные рассуждения, определим точки t_1, \dots, t_N следующим образом

$$t_j = \delta(j - 2L) + \min_{i=1, \dots, m} x_i,$$

$$j = 1, \dots, N, \quad N = 2L_{\max} + 4L.$$

Отметим, что особенности анализируемых рядов (а именно, монотонность и медленное изменение) позволяют нам рассматривать только несколько ведущих компонент в (1.12) для сглаживания. В отличие от задач извлечения сигнала с фиксированным L и оценкой r , мы фиксируем r и регулируем L для контроля гладкости.

Для наглядности мы сейчас продемонстрируем влияние параметра L на оценку SSA $\hat{\mathcal{F}}_m(L, r)$ при $r = 1$ наглядно.

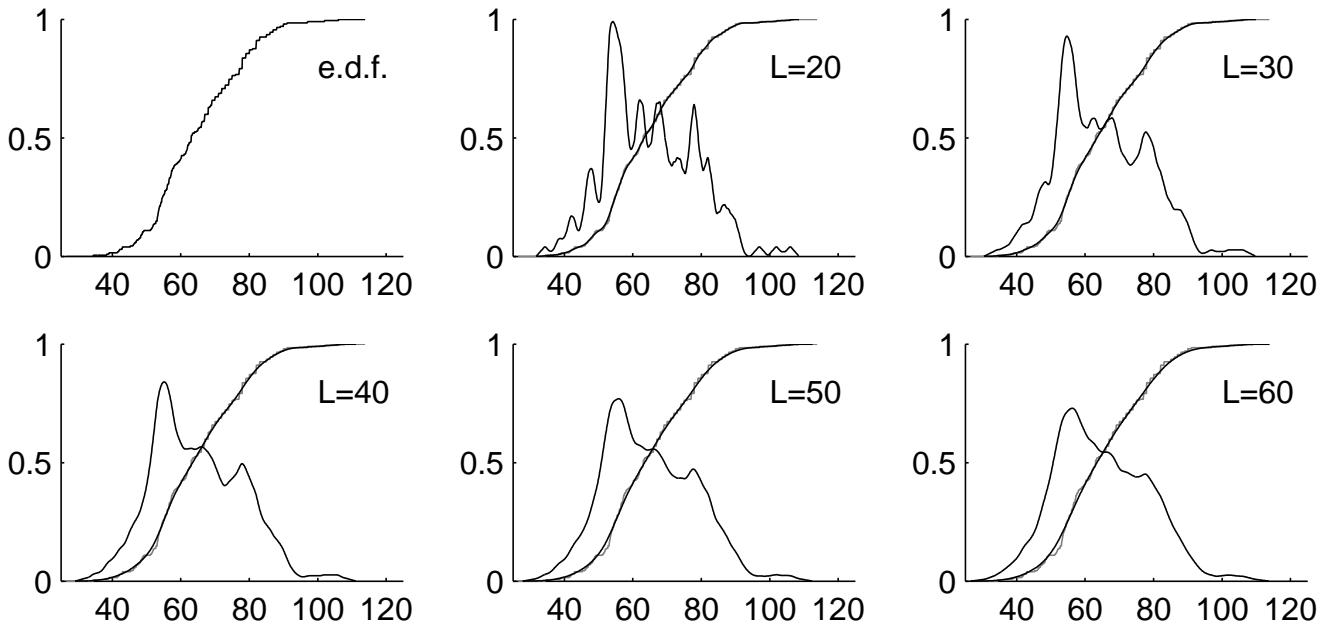


Рис. 7.5: Эмпирическая функция распределения для измерения сухой массы тела по данным Австралийского института спорта и оценки SSA функции распределения и плотности для $L = 20, 30, \dots, 60$.

Пример 4. Рассмотрим 202 измерения сухой массы тела из Австралийского института спорта, которые были изучены, в частности, в [194]. На рис. 7.5 изображен эмпирическая функция распределения для этих измерений (на основе которых получаем $\delta = 0.13$ и $L_{\max} = 274$) и оценки SSA для $L = 20, 30, \dots, 60$. Плотности на рисунке перенасштабированы для удобства визуального восприятия. Видно, что SSA оценка $\hat{F}_m(L, 1)$ становится более гладкой и число мод соответствующей плотности уменьшается с увеличением параметра L . Сравнивая с ядерным сглаживанием, видим, что малые значения L соответствуют малым полосам пропускания, а большие значения L — большим полосам пропускания, и оба метода приводят к относительно схожим результатам для данного набора данных.

Пример 5. В качестве второго примера мы рассмотрим 22 наблюдения кремнезема в хондритовых метеоритах [195]. На рис. 7.6 мы изобразили эмпирическую функцию распределения для этих наблюдений (которые дают $\delta = 0.043$ и $L_{\max} = 163$) и оценки SSA для $L = 20, 40, \dots, 100$. Обратим внимание, что плотность имеет три моды для большого диапазона значений L . В общем, оценка SSA может быть использована для исследования мультиомодальности аналогично тому, как это обсуждалось в [193].

Специальные SSA оценки и коррекция смещения

Стоит изучить оценку SSA для выбранных значений r , числа ведущих компонент, используемых в (7.1), более подробно и разработать процедуру для обеспечения монотонности окончательной оценки.

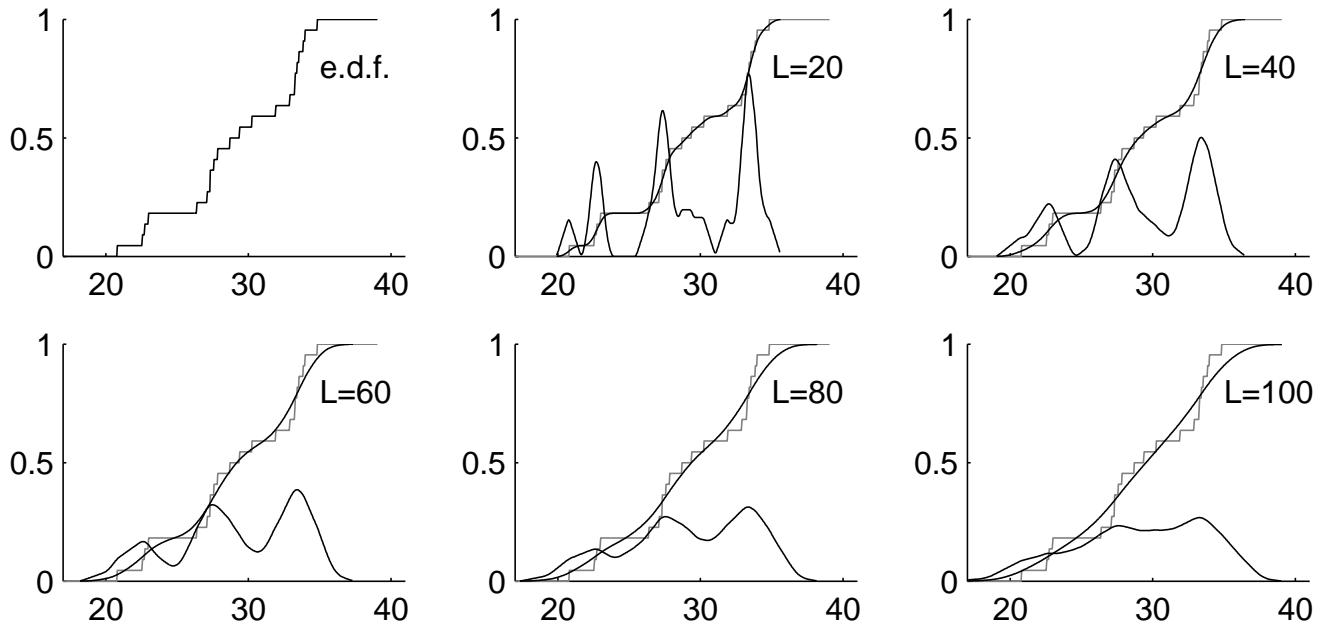


Рис. 7.6: Эмпирическая функция распределения для наблюдений хондитовых метеоров и оценки SSA функции распределения и плотности для $L = 20, 40, \dots, 100$.

Оценка SSA, как и любая процедура сглаживания, страдает от погрешности оценки. Поэтому мы предлагаем эффективный подход для коррекции смещения, где смещение оценивается путем повторного применения фильтра SSA. Имитационное исследование оценивает эффективность оценок SSA с $r = 1$ или $r = 2$ и версии с коррекцией смещения в терминах трех стандартных мер, включая интегрированное среднеквадратичное отклонение и расстояние Колмогорова-Смирнова.

Оценка SSA при $r = 1$ Случай $r = 1$ дает простейшую оценку, которая будет обозначаться как оценка SSA^{1c} , поскольку используется одна компонента SVD-разложения (1.12). В следующей лемме мы устанавливаем, что алгоритм SSA при $r = 1$ дает достоверный результат.

Лемма 18. *Оценка SSA^{1c} (7.2) является допустимой функцией распределения.*

Доказательство. Поскольку элементы ряда \mathcal{F}_m положительны, элементы $\mathbb{X}\mathbb{X}^T$ также положительны. Из этого следует, что элементы ведущего собственного вектора U_1 имеют одинаковый знак, который можно считать положительным. Поэтому монотонность оценки SSA^{1c} следует из (7.3).

Применяя неравенство Йенсена, получаем

$$\sum_{i=1}^L \sum_{l=1}^L u_{1,i} u_{1,l} / L = \left(\sum_{i=1}^L u_{1,i} \right)^2 / L \leq \sum_{i=1}^L u_{1,i}^2 = 1.$$

Поэтому, используя (7.3), имеем $0 \leq \hat{f}_j \leq 1$, что завершает доказательство. \square

Оценка SSA при $r = 2$ Отметим, что оценка $\hat{\mathcal{F}}_m(L, 2)$ (т.е. оценка SSA с $r = 2$) может представлять собой ряд, который не обязательно является дискретной версией функции распределения. В частности, ряд $\hat{\mathcal{F}}_m(L, 2)$ может быть немонотонным. Введем оператор $M(\mathcal{H})$ для заданного неубывающего ряда $\mathcal{H} = (H_1, \dots, H_N)$, $H_i \in \mathbb{R}$, $1 \leq i \leq N$, такой, что $M(\mathcal{H})$ является дискретной версией функции распределения и близка к \mathcal{H} . Этого можно достичь, заменив все отрицательные значения на нули и все значения превышающие единицу, на единицы, а затем взять среднее лево- и правомонотонных функций. Таким образом, мы определяем оценку SSA^{2c} как $\hat{\mathcal{F}}_m^{2c}(L) = M(\hat{\mathcal{F}}_m(L, 2))$.

Наши численные результаты (см. таблицу 7.6) показывают, что оценка SSA при $r > 2$ в среднем не является действительно лучшей. Это можно объяснить следующим образом. В теории SSA одной из целей SSA является извлечение ряда конечного ранга τ из зашумленного ряда, где ряд конечного ранга τ — это ряд, чья траекторная матрица \mathbb{X} имеет ровно τ ненулевых собственных значений. Поскольку функции распределения обычно не имеют конечного ранга, возникает проблема аппроксимации ряда \mathcal{F}^* рядом конечного ранга. Точность этого приближения зависит от L и r , которые совместно дают слишком большую свободу. Поэтому для практических целей обычно достаточно оценки SSA с $r = 2$.

Оценка SSA с $r = 1$ и коррекция смещения Напомним, что оценка SSA^{1c} представляет собой линейный фильтр размера $2L - 1$. Более того, доказательство леммы 1 показывает, что этот фильтр является взвешенным скользящим средним с положительными коэффициентами. Поэтому оценка SSA^{1c} в общем случае имеет смещение, которое желательно устранить.

Поэтому мы вводим оператор S , определенный как оценка SSA^{1c}, то есть

$$S(\mathcal{F}_m|L) = \hat{\mathcal{F}}_m(L, 1).$$

Мы также рассматриваем оператор S^k , который означает многократное применение оператора S , то есть $S^k = S(S^{k-1})$. Заметим, что $S^k(\mathcal{F}|L)$ означает, что фильтр SSA^{1c} последовательно применяется k раз. Чтобы вычислить $S^k(\mathcal{F}|L)$, ряд \mathcal{F} должен иметь по крайней мере $(1+k)L$ нулей в начале и $(1+k)L$ единиц в конце. Для дальнейших рассуждений мы предполагаем, что m достаточно велико, так что $S(\mathcal{F}^*|L) \approx S(\mathcal{F}_m|L)$, и L относительно мало.

Определим оператор

$$S^b(\mathcal{F}_m|L) = 3S(\mathcal{F}_m|L) - 3S^2(\mathcal{F}_m|L) + S^3(\mathcal{F}_m|L),$$

который имеет меньшее смещение, чем оператор S , как доказано в разделе 7.2.2. Заметим, что ряд $S^b(\mathcal{F}|L)$ не обязательно монотонный, но он меняется от 0 к 1. Поэтому ряд $\hat{\mathcal{F}}_m^b(L) = M(S^b(\mathcal{F}_m|L))$ дает оценку, которую будем называть оценкой SSA^b для \mathcal{F}^* .

7.2.2. Качество оценок SSA

Чтобы получить некоторое представление о свойствах конечной выборки, мы исследуем оценки SSA с помощью моделирований рядов и сравним их с ядерной оценкой плотности с полосой пропускания, оцененной с помощью кросс-валидации по методу наименьших квадратов (LSCV).

Для этого мы рассматриваем интегрированную квадратичную ошибку (ISE)

$$D_{\text{ISE}}(\hat{p}) = \int (\hat{p}(x) - p(x))^2 dx,$$

расстояние Колмогорова-Смирнова

$$D_{\text{KS}}(\hat{F}) = \|\hat{F} - F\|_{\infty} = \max_x |\hat{F}(x) - F(x)|$$

и расстояние Хеллингера

$$D_{\text{H}}(\hat{p}) = \int (\sqrt{\hat{p}(x)} - \sqrt{p(x)})^2 dx,$$

где $p(x)$ — плотность, а $F(x)$ — функция распределения для предполагаемой модели.

Мы рассматриваем две модели: нормальное распределение $N(0, 1)$ и смесь $0.4N(0, 1) + 0.6N(5, 2^2)$, и вычисляем средние значения вышеуказанных расстояний, используя 10000 сгенерированных выборок размера 100. Эти настройки дают $\delta \approx 0.01$ и $L_{\max} \approx 240$ для первой модели и $\delta \approx 0.03$ и $L_{\max} \approx 200$ для второй модели.

На рис. 7.7 представлены ядерные оценки плотности и оценки SSA с $L = 100$ и $L = 130$ для пяти выборок размером 100 из $N(0, 1)$, в то время как эти оценки с $L = 60$ и $L = 70$ для смеси $0.4N(0, 1) + 0.6N(5, 2^2)$ изображены на рис. 7.8. Мы можем заметить, что оценки SSA немного более гладкие, чем ядерные оценки плотности с полосой пропускания LSCV.

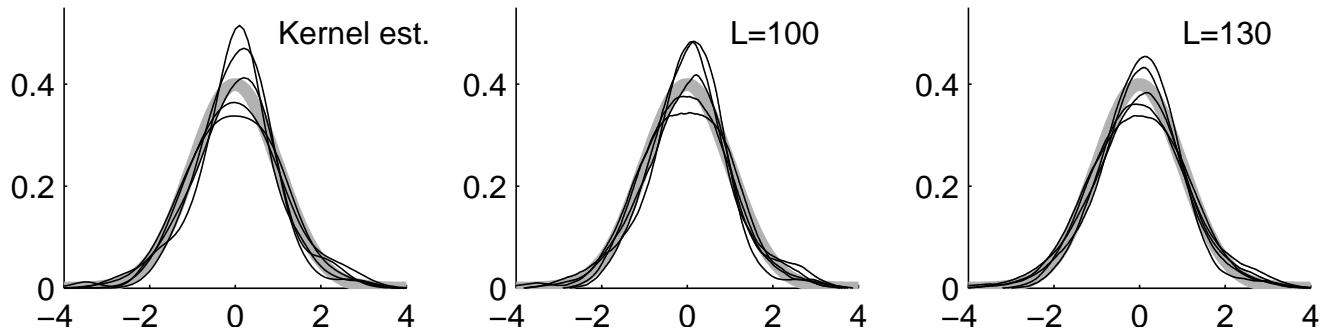


Рис. 7.7: Ядерные оценки плотности и оценки SSA^{1c} с $L = 100$ и $L = 130$ для выборок размером 100 из $N(0, 1)$, плотность которых выделена серым цветом.

В таблице 7.5 приведены средние значения ISE, расстояния Колмогорова-Смирнова и Хеллингера оценок до истинных функций распределения. Мы видим, что оценки SSA^{1c},

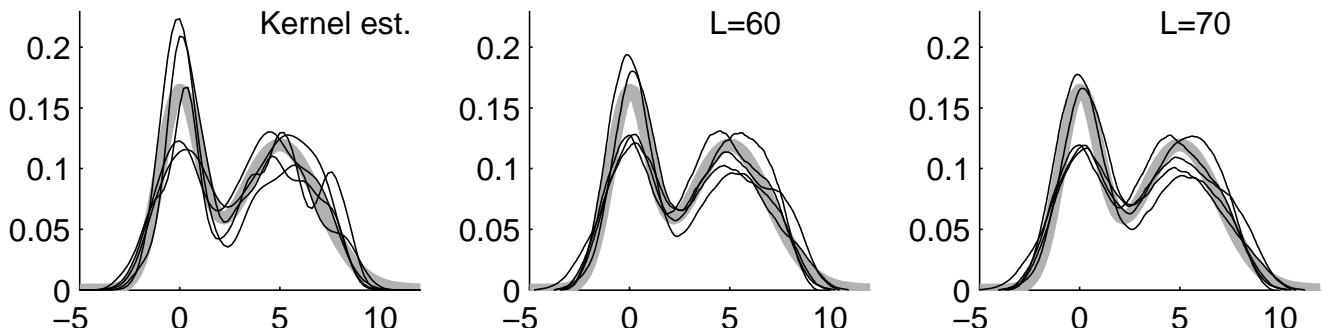


Рис. 7.8: Ядерные оценки плотности и оценки SSA^{1c} с $L = 60$ и $L = 70$ для выборок размером 100 из $0.4N(0, 1) + 0.6N(5, 2^2)$, плотность которых выделена серым цветом.

SSA^{2c} и SSA^b с L из широкого диапазона показывают лучшие результаты, чем ядерные оценки плотности, по всем трем критериям.

Таблица 7.5: Средние значения ISE, расстояния Колмогорова-Смирнова и Хеллингера для ядерных оценок, оценки плотности SSA^{1c} , SSA^{2c} и SSA^b для выборок размером 100.

	ED_{ISE}	ED_{KS}	ED_H
модель $N(0, 1)$			
Ядерная оценка с h_{LSCV}	0.0071	0.0551	0.0143
Оценка SSA^{1c} с $L=80$	0.0065	0.0521	0.0122
Оценка SSA^{1c} с $L=100$	0.0055	0.0509	0.0113
Оценка SSA^{1c} с $L=130$	0.0061	0.0550	0.0139
Оценка SSA^{2c} с $L=160$	0.0054	0.0496	0.0135
Оценка SSA^{2c} с $L=190$	0.0048	0.0486	0.0122
Оценка SSA^{2c} с $L=220$	0.0048	0.0493	0.0117
Оценка SSA^b с $L=130$	0.0054	0.0497	0.0144
Оценка SSA^b с $L=160$	0.0046	0.0479	0.0135
Оценка SSA^b с $L=190$	0.0048	0.0492	0.0137
модель $0.4N(0, 1) + 0.6N(5, 2^2)$			
Ядерная оценка с h_{LSCV}	0.0058	0.0617	0.0231
Оценка SSA^{1c} с $L=50$	0.0044	0.0595	0.0172
Оценка SSA^{1c} с $L=60$	0.0044	0.0593	0.0179
Оценка SSA^{1c} с $L=70$	0.0048	0.0603	0.0204
Оценка SSA^{2c} с $L=80$	0.0045	0.0598	0.0190
Оценка SSA^{2c} с $L=90$	0.0042	0.0591	0.0179
Оценка SSA^{2c} с $L=100$	0.0042	0.0588	0.0174
Оценка SSA^b с $L=70$	0.0044	0.0608	0.0193
Оценка SSA^b с $L=80$	0.0042	0.0602	0.0182
Оценка SSA^b с $L=90$	0.0044	0.0607	0.0185

Мы заметили, что большие значения L используются для модели, заданной нормальным распределением, а меньшие значения L используются для модели с двумя модами. Это похоже на поведение полосы пропускания LSCV, которая уменьшается по мере того, как

распределение становится менее нормальным.

Мы можем заметить, что каждая оценка SSA имеет свой собственный диапазон подходящих значений L . Оценку SSA^b следует использовать при больших значениях L , так как это дает большую гладкость, в то время как оценка SSA^{1c} для таких L имеет заметное смещение. Между тем, диапазоны для оценок SSA^{2c} и SSA^b относительно близки.

Автоматическая процедура выбора параметра сглаживания предложена соавтором Андреем Пепельшевым в [7], поэтому мы ссылаемся на эту работу и не помещаем этот раздел здесь.

Обоснование аппроксимации смещения

Рассмотрим оператор свертки $C(F) = \mu_\varepsilon * F$, где μ_ε — мера с конечными моментами,

$$C(F)(x) = \int F(x-y)\mu_\varepsilon(dy),$$

и мы интерпретируем μ_ε как ядро с шириной ε . Если μ_ε — дискретная мера, то $C(F)$ — это взвешенное скользящее среднее значений функции F .

Пусть нашей задачей является оценка функции F с помощью линейных комбинаций $C^j(F)$ для $j = 1, 2, \dots$, где C^j — j -кратная композиция оператора C , $C^0(F) = F$. Поэтому мы рассматриваем оценку F в виде $\widehat{F}_P(\mu_\varepsilon) = P(\mu_\varepsilon) * F$, где P — многочлен без свободного члена. Заметим, что эта оценка имеет требуемый вид, так как $P(0) = 0$. Кроме того, ошибка оценки равна $P(\mu_\varepsilon) * F - F = P^{\text{err}}(\mu_\varepsilon) * F$, где $P^{\text{err}} = P - 1$.

Пусть оценка $\widehat{F}_P(\mu_\varepsilon)$ порождена $P_k(z) = 1 - (1-z)^k$. Тогда имеем $P_k^{\text{err}}(z) = -(1-z)^k$. Чтобы изучить эту оценку, мы определим $w_{k,\varepsilon} = \int y^k \mu_\varepsilon(dy)$ для $k \in \mathbb{N}$, $w_{0,\varepsilon} = \int \mu_\varepsilon(dy) - 1$ и рассмотрим формальное разложение Тейлора гладкой функции F в точке x :

$$F(x-y) = \sum_{i=0}^{\infty} (-1)^i \frac{F^{(i)}(x)}{i!} y^i$$

и его производные

$$F^{(j)}(x-y) = \sum_{i=0}^{\infty} (-1)^i \frac{F^{(i+j)}(x)}{i!} y^i.$$

Тогда мы имеем следующее формальное разложение члена ошибки:

$$\widehat{F}_{P_k}(\mu_\varepsilon) - F = P_k^{\text{err}}(\mu_\varepsilon) * F = \sum_{i_1, \dots, i_k=0}^{\infty} (-1)^{k+1+\sum i_j} \frac{F^{(\sum_{j=1}^k i_j)}(x)}{\prod_{j=1}^k i_j!} \prod_{j=1}^k w_{i_j, \varepsilon}. \quad (7.5)$$

Заметим, что оценка SSA^{1c} соответствует случаю $k = 1$, где $P_1(z) = 1 - (1-z) = z$ подразумевает $\widehat{F}_{P_1}(\mu_\varepsilon) = \mu_\varepsilon * F = C(F)$. Между тем, оценка SSA^b соответствует случаю $k = 3$, где $P_3(z) = 1 - (1-z)^3 = z^3 - 3z^2 + 3z$ влечет за собой $\widehat{F}_{P_1}(\mu_\varepsilon) = C^3(F) - 3C^2(F) + C(F)$.

Формула (7.5) показывает, что если $w_{j,\varepsilon} \rightarrow 0$ при $\varepsilon \rightarrow 0$ с достаточной скоростью сходимости (например, $w_{j,\varepsilon} = O(\varepsilon^j)$), тогда ошибка $\widehat{F}_{P_k}(\mu_\varepsilon) - F$ уменьшается с ростом k при малых ε .

Напомним, что оператор S можно представить как оператор C с мерой μ_ε , сосредоточенной в точках $\pm i\delta$, $i = 0, \dots, L$. Поэтому μ_ε близка к дельта-функции Дирака, сосредоточенной в 0, если L мала. Таким образом, можно ожидать, что оценка SSA^b достаточно гладкой функции распределения F имеет меньшее смещение, чем оценка SSA^{1c} , что подтверждается обширными численными исследованиями.

Исследование с помощью моделирования

В таблице 7.6 мы представляем эффективность автоматической процедуры для оценок по SSA и ядерных оценок плотности. Мы видим, что полоса пропускания Sheather-Jones plug-in (SJPI) лучше, чем полоса пропускания LSCV, а полоса пропускания Indirect cross-validation (ICV) не совсем хороша для модели нормального распределения. Однако полоса пропускания SJPI хуже, чем полоса пропускания LSCV, если модель имеет две моды, значительно отличающиеся по ширине. Стоит отметить, что плотность с полосой пропускания LSCV дает небольшой ISE для модели $0.3N(0,1) + 0.7N(15,4^2)$, но дает большое количество мод, в то время как плотность с другими полосами пропускания и оценки SSA имеют обычно 2 или 3 моды.

Можно заметить, что оценки SSA^{1c} , SSA^b и SSA^{2c} с автоматическим выбором параметра сглаживания дают меньшие ошибки ISE по сравнению с ядерной оценкой плотности, в то время как результаты для расстояний Колмогорова-Смирнова и Хеллингера похожи или немного лучше. Оценка SSA^{3c} полезна только для распределений с тяжелыми хвостами, которые можно определить по различию значений L_a для $r = 1, 2, 3$. В целом, оценка SSA^{2c} имеет лучшую эффективность.

7.2.3. Приложение оценки функции распределения при исследовании рынка продаж

В данном разделе мы анализируем реальный набор данных из маркетингового исследования, где в результате огрубления данных оценка квантилей является сложной задачей.

В маркетинговых исследованиях наблюдаемые значения обычно огрублываются. Например, в анкетах возраст должен быть заполнен целыми числами, в то время как запрашиваемые цены представляют собой смесь значений, округленных до целых чисел и кратных десяти, а заявленные доходы округлены до сотен или тысяч [196]. В этих случаях вычисление квантилей через порядковую статистику не дает хороших результатов. Таким образом, крайне важно получить сглаженное распределение и затем вычислить квантили с его помощью.

В качестве примера рассмотрим один признак из исследования цен на продукцию. В

Таблица 7.6: Средние значения ISE, расстояния Колмогорова-Смирнова и Хеллингера для ядерных оценок плотности с различными полосами пропускания и оценки SSA^{1c} , SSA^b и SSA^{2c} с использованием автоматической процедуры для выборок размером 100. В последнем столбце приведено среднее значение параметра L_a , которое найдено автоматической процедурой

	\mathbf{ED}_{ISE}	\mathbf{ED}_{KS}	\mathbf{ED}_H	\mathbf{EL}_a
модель $N(0, 1)$				
Ядерная оценка с h_{LSCV}	0.0071	0.0551	0.0143	
Ядерная оценка с h_{SJPI}	0.0066	0.0536	0.0131	
Ядерная оценка с h_{ICV}	0.0075	0.0546	0.0146	
Оценка SSA^{1c}	0.0061	0.0537	0.0128	107.6
Оценка SSA^{2c}	0.0060	0.0503	0.0142	145.7
Оценка SSA^{3c}	0.0090	0.0556	0.0189	143.4
Оценка SSA^b	0.0052	0.0488	0.0141	141.7
модель $0.4N(0, 1) + 0.6N(5, 2^2)$				
Ядерная оценка с h_{LSCV}	0.0058	0.0617	0.0231	
Ядерная оценка с h_{SJPI}	0.0052	0.0609	0.0210	
Ядерная оценка с h_{ICV}	0.0055	0.0614	0.0229	
Оценка SSA^{1c}	0.0051	0.0607	0.0215	67.6
Оценка SSA^{2c}	0.0047	0.0610	0.0195	86.4
Оценка SSA^{3c}	0.0054	0.0623	0.0219	90.3
Оценка SSA^b	0.0052	0.0617	0.0206	89.7
модель $0.3N(0, 1) + 0.7N(15, 4^2)$				
Ядерная оценка с h_{LSCV}	0.0048	0.0672	0.0394	
Ядерная оценка с h_{SJPI}	0.0069	0.0733	0.0602	
Ядерная оценка с h_{ICV}	0.0049	0.0670	0.0396	
Оценка SSA^{1c}	0.0053	0.0679	0.0451	40.9
Оценка SSA^{2c}	0.0046	0.0660	0.0380	60.0
Оценка SSA^{3c}	0.0043	0.0643	0.0349	73.8
Оценка SSA^b	0.0047	0.0670	0.0391	48.9

частности, пусть x_i — это цена билета, такая, что i -ый респондент купит билет по этой цене, $i = 1, \dots, 475$.

На рис. 7.9 мы изобразили график частот выборки, эмпирическую функцию распределения и оценку SSA^{2c} с использованием автоматической процедуры для накопленных данных. Видно, что большинство респондентов округлили цены до кратных десяти. Оценки медианы и квантилей, которые представляют интерес для менеджеров по маркетингу, $\tilde{q}_{1/5} = 60$, $\tilde{q}_{1/2} = 70$ и $\tilde{q}_{4/5} = 80$ были вычислены через порядковые статистики. Однако, используя SSA^{2c} оценку функции распределения, мы получаем $\hat{q}_{1/5} = 57.27$, $\hat{q}_{1/2} = 71.99$ и $\hat{q}_{4/5} = 82.14$, которые являются более точными значениями.

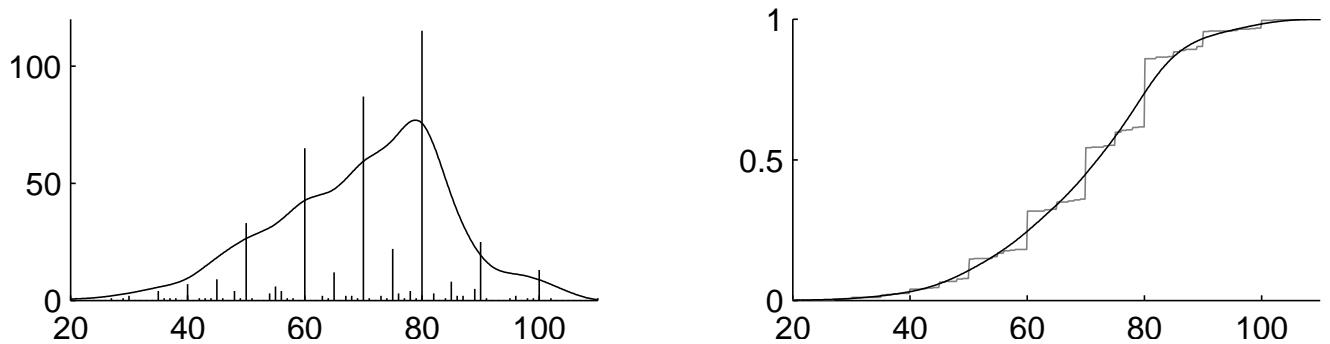


Рис. 7.9: Слева: Частотный график цен на билеты и оценка плотности SSA^{2c} с использованием автоматической процедуры для накопленных данных. Справа: Эмпирическая и сглаженная функции распределения для цены билета

7.3. Двухэкспоненциальные модели паттернов экспрессии генов для зашумленных экспериментальных данных

Формирование пространственного паттерна первичного передне-заднего морфогенетического градиента транскрипционного фактора *Bicoid* (*Bcd*) изучалось экспериментально и вычислительно в течение многих лет [197]. *Bcd* определяет позиционную информацию для последующих генов сегментации, влияя на план тела муши. В последнее время ряд исследователей сосредоточились на динамике паттерна основного градиента *Bcd* мРНК, которая транслируется в белок *Bcd*. Новые, более точные методы визуализации *Bcd* мРНК должны сочетаться с методами количественного извлечения сигнала для восстановления распределения *Bcd* мРНК.

Здесь мы представляем надежный метод количественной оценки градиентов с помощью двухэкспоненциальной модели. Этот подход: 1) имеет естественные, биологически значимые параметры; и 2) инвариантен к линейным преобразованиям данных, которые могут возникнуть из-за вариаций в экспериментальных условиях (например, настройки микроскопа, неспецифический фоновый сигнал). Это позволяет нам количественно оценить *Bcd* изменчивость градиента мРНК от эмбриона к эмбриону (что важно для изучения устойчивости регуляторных сетей развития), отсортировать нетипичные градиенты и классифицировать эмбрионы по стадиям развития по количественным параметрам градиента. Результаты этого раздела следуют [17] и [21].

Подход. При визуализации целых эмбрионов вариабельность может возникать во время фиксации тканей и окрашивания флуорофорами, а также из-за различий в настройках микроскопа (усиление и смещение) между измерениями разных партий эмбрионов в разные дни. Здесь мы обсуждаем особенности извлечения нужных данных методом, который не чувствителен к таким экспериментальным вариациям.

Цель нашего подхода — создать модель для базального и апикального профилей (см.

рис. 7.10В) для градиента *Bcd*, оценить параметры модели и показать, что они могут помочь получить биологические результаты; в частности, сравнить различные возрасты в развитии эмбриона. Мы показываем пример того, как данные, полученные и смоделированные с помощью этой техники, могут дать новое биологическое понимание *Bcd* образования градиента РНК.

Новизна подхода заключается в учете параметров модели, которые не зависят от линейного преобразования данных и тем самым от неспецифического фонового сигнала и настроек микроскопа. Это очень важно, поскольку в противном случае результаты сравнения могут быть обусловлены условиями эксперимента, а не биологическими причинами.

Модель. Двухэкспоненциальная модель паттерна профиля белка *Bcd* может быть хорошо аппроксимирована одной экспонентой плюс почти постоянный фон [198, 45]. В отличие от этого, в то время как некоторые *Bcd* РНК-профили демонстрируют такое поведение, многие другие, особенно на ранних стадиях, показывают гораздо более резкий экспоненциальный спад в передней части, плюс постоянный или даже задне-возрастающий компонент в остальной части эмбриона (рис. 7.11). Переход между компонентами может быть легко заметен в моделях РНК (но не в белке), как “перегиб” около позиции 20-30% длины яйца (%EL). Эти различные компоненты предполагают несколько масштабов (или механизмов) в переносе *Bcd* РНК в заднем направлении.

Техника. В предыдущих работах мы применяли метод извлечения сигнала, основанный на сингулярном спектральном анализе (SSA), для количественной оценки градиентов белка *Bcd* [45]. Это продемонстрировало, что SSA может надежно и автоматически извлекать градиенты белка AP *Bcd*. Они представляли собой сумму двух экспонент, одна со значительной константой затухания (сильная кривизна) и одна почти линейной формы, отражающая неспецифический фоновый сигнал. Здесь мы адаптируем метод SSA к более сложным случаям *Bcd* градиентов РНК, подтверждая надежность и эффективность подхода. Сам SSA используется для выделения сигнала, а связанный с SSA метод ESPRIT [108, 29] используется для оценки параметров сигнала. Методы SSA доказали свою устойчивость к извлечению сигнала из данных со значительной экспериментальной изменчивостью и собственным шумом [1, 199, 45, 29].

7.3.1. Методы

Двухэкспоненциальное моделирование

Описание модели Мы рассматриваем следующую двухэкспоненциальную функцию (от расстояния по антериор-постериор оси AP, координата x) как модель паттерна *Bcd* мРНК, чтобы отразить отчетливый двухкомпонентный паттерн большинства *Bcd* градиентов РНК (с “перегибом”, обычно наблюдаемым при 20–30 %EL):

$$s(x) = C_1 \exp(\alpha_1 x) + C_2 \exp(\alpha_2 x), \quad (7.6)$$

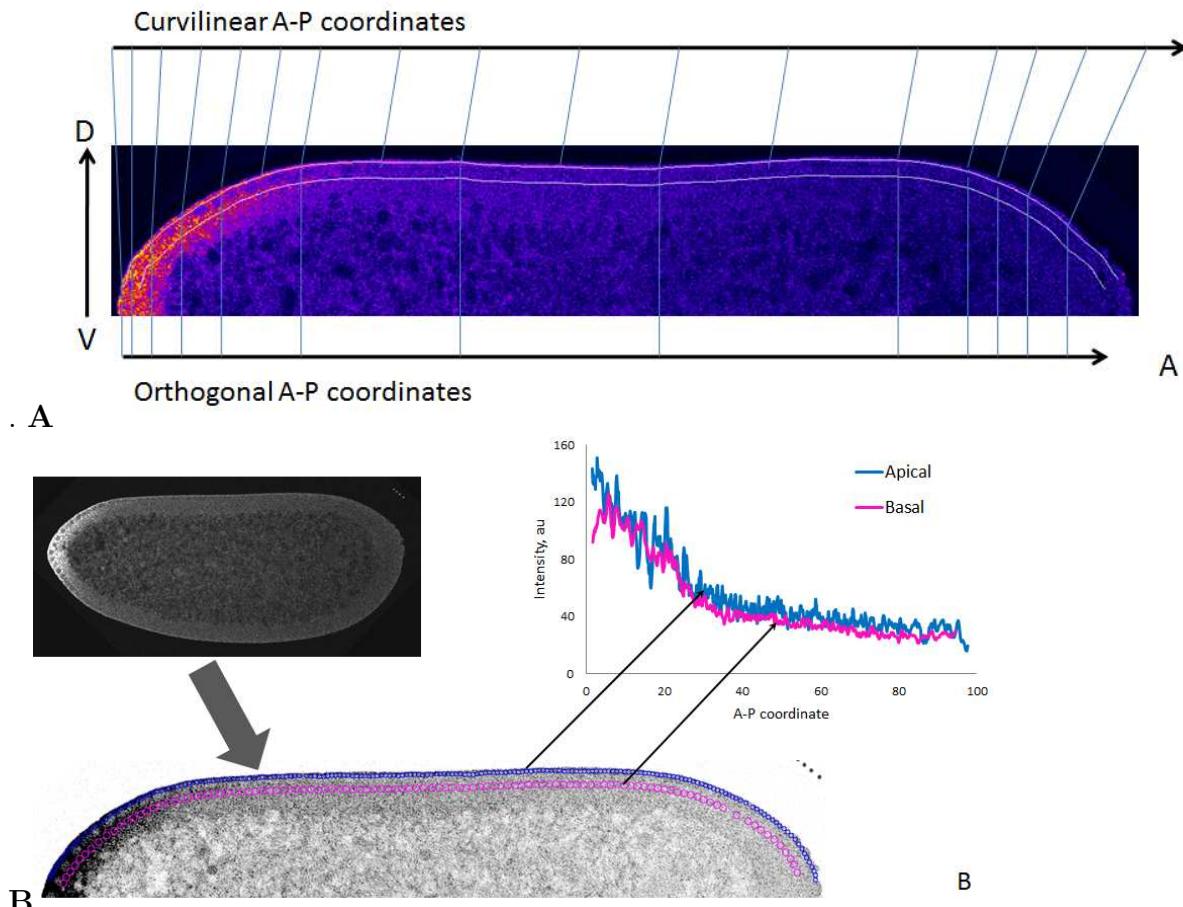


Рис. 7.10: Подготовка данных для количественного анализа сагиттальных изображений методом 1D Singular Spectrum Analysis (SSA). **A.** Интенсивность флуоресценции пропорциональна концентрации *Bcd* mRNA. Градиент *Bcd* mRNA в основном направлен от головы к хвосту, AP, (слева направо), но можно увидеть DV вариации (координаты сверху вниз), а также вариации по глубине эмбриона (направление BA). Для транспортировки и паттерна вдоль поверхности эмбриона естественные координаты являются криволинейными. Для выделения градиентного паттерна от головы к хвосту криволинейные координаты хорошо аппроксимируются проекцией на ось AP. **B.** Для количественной оценки градиента по AP и различий в BA мы отбираем данные из апикального слоя над кортикальными ядрами и базального слоя под кортикальными ядрами, используя цепочки перекрывающихся областей интереса (ROIs). Данные из каждого слоя анализируются независимо с помощью 1D SSA. Затем каждый слой может быть изображен в виде графика зависимости интенсивности от положения AP (рисунок справа вверху).

или

$$s(x) = C_1 \lambda_1^x + C_2 \lambda_2^x,$$

для $\lambda_i = e_i^\alpha$. Эти два компонента, первый из которых мы называем *anterior* (для резкого, быстро затухающего паттерна в передней части), а второй — *shallow* (для более близкому к постоянному компоненту в средней и задней части эмбриона), имеют по два параметра — амплитуду C и скорость убывания α (или основание экспоненты λ). В биологических терминах, C в $C\lambda^x$ для $\lambda < 1$ представляет максимальную концентрацию экспоненциального компонен-

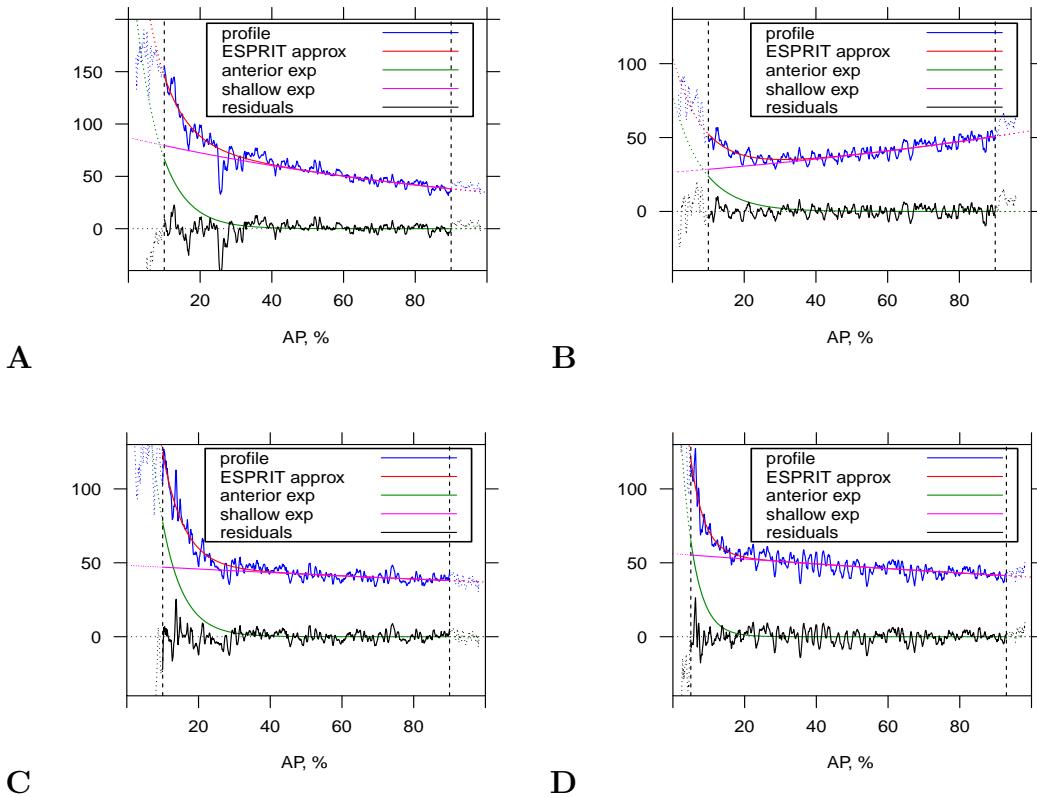


Рис. 7.11: Репрезентативные примеры профилей AP *Bcd* мРНК, иллюстрирующие разнообразие случаев и эффективность подхода моделирования. Исходные данные зашумлены; гладкий паттерн представляет собой подгонку методом ESPRIT, которая является суммой двух экспонент; они изображены по отдельности (уменьшающаяся до нуля передняя и мелкая хвостовая; последняя похожа на прямую линию). Остаточный шум колеблется вокруг нуля. **A.** Эмбрион раннего цикла ядерного расщепления Cleavage 14A (nc14) с типичной не быстро убывающей передней экспонентой и медленно убывающим 2-м компонентом, распространяющимся по всему эмбриону (см. [200]). **B.** Профиль *Bcd* мРНК, в котором 2-й, магистральный, компонент поднимается к задней части (т.е. имеет положительный экспоненциальную показатель). **C.** Случай с почти плоским 2-м компонентом (представляющим сигнал мРНК после 25%EL). **D.** Эмбрион с очень резкой передней (1-й компонент) экспонентой, падающей до низких значений к 10%EL.

та, а α (и, следовательно, λ) представляет скорость, с которой компонент уменьшается (или увеличивается) вдоль координаты AP. Передняя экспонента всегда уменьшается, поэтому $\lambda_1 < 1$, $\alpha_1 < 0$; в то время как малая экспонента может уменьшаться или увеличиваться (рис. 7.11).

Модель одной экспоненты плюс постоянного фона является частным случаем уравнения (7.6), с показателем $\alpha_2 = 0$ (основанием $\lambda_2 = 1$). Использование (7.6) не требует наличия двух сильных (ненулевых α) экспонент в сигнале (паттерне). В случае модели, обычно применяемой к градиенту белка Bcd, первая экспонента описывает сигнал, а вторая экспонента описывает неспецифический фоновый сигнал и смещение микроскопа.

Обратим внимание, что необработанные данные изображения, скорее всего, имеют вид $s(x) + \varepsilon(x)$, где $\varepsilon(x)$ представляет собой “шум”, т.е. нерегулярные колебания с нулевым средним.

Характеристики модели не зависят от усиления/смещения микроскопа и фона
Для устранения влияния непостоянства настроек микроскопа (усиление и смещение) и неизвестной формы неспецифического фона [198, 201, 202], могут использоваться только характеристики градиента, которые не изменяются при линейном преобразовании градиента.

То есть, если каждый профиль (градиент) может быть представлен линейным преобразованием:

$$f(x) = A(s(x) + \varepsilon(x)) + B, \quad (7.7)$$

где $A > 0$ и B представляют собой неизвестный масштаб и неизвестное смещение, соответственно. Вероятно, A и B будут отличаться у разных эмбрионов (при различных условиях окрашивания, настройках микроскопа и т.д.), но когда мы берем базальные и апикальные профили в пределах каждого изображения эмбриона, мы предполагаем, что A и B постоянны в пределах одного эмбриона. Для сравнения данных между эмбрионами мы используем независимость характеристик профиля от линейных преобразований, т.е. независимость от значений A и B .

Сигнал (7.6) имеет характеристики, которые приближенно удовлетворяют независимости от линейных преобразований, если второй (мелкоградиентный) экспоненциальный показатель достаточно мал ($\alpha_2 \approx 0$, $\lambda_2 = e^{\alpha_2} \approx 1$) и поэтому может быть аппроксимирована линейной функцией. Это разумное предположение для данных *Bcd* мРНК, дающее

$$\begin{aligned} A(C_1 \exp(\alpha_1 x) + C_2 \exp(\alpha_2 x)) + B \approx \\ AC_1 \exp(\alpha_1 x) + AC_2(1 + \alpha_2 x) + B \approx \tilde{C}_1 \exp(\tilde{\alpha}_1 x) + \tilde{C}_2 \exp(\tilde{\alpha}_2 x), \end{aligned}$$

где

$$\tilde{C}_1 = AC_1, \quad \tilde{\alpha}_1 = \alpha_1, \quad \tilde{C}_2 = AC_2 + B, \quad \tilde{\alpha}_2 = AC_2 \alpha_2 / (AC_2 + B).$$

Видно, что некоторые параметры (например, предэкспоненциальные коэффициенты C_1 и C_2) зависят от A и/или B , в то время как показатель первой экспоненты α_1 не зависит от линейных преобразований. Кроме того, мы предполагаем, что коэффициенты перед первой и второй экспонентами положительны (что верно, в общем случае, для данных *Bcd* РНК) при $A > 0$ и B , которые фигурируют в экспериментах. Затем, знак α_2 также устойчив к преобразованию и определяет, увеличивается или уменьшается вторая (малая) экспонента. Отрицательный знак C_1 или C_2 может быть использован для обнаружения профилей, которые не соответствуют модели.

Заметим, что аналогичные соображения о параметрах, не зависящих от линейных пре-

образований, можно применить к любому числу активных экспонент (например, можно рассмотреть как переднюю, так и хвостовую и изучить их показатели), в то время как малая фоновая экспонента должна быть одна. Важно, что уровень шума напрямую зависит от масштабирования A , поэтому мы не используем его в качестве характеристики эмбриона.

Модельные характеристики апикального и базального профилей Мы применяем модель как к апикальным, так и к базальным профилям, добавляя соответствующие верхние индексы к обозначениям параметров. Несколько характеристик, которые отражают связь между двумя профилями одного и того же эмбриона (поэтому A и B одинаковы для обоих профилей), могут быть добавлены. Для ясности мы будем писать ‘*anterior*’ или ‘*shallow*’ в нижних индексах вместо 1 и 2 соответственно.

Таким образом, следующие комбинации модельных параметров профилей можно считать практически независимыми от линейного преобразования интенсивностей, т.е. от A и B :

$$\alpha_{\text{anterior}}^{(\text{apical})}, \quad \text{sign}(\alpha_{\text{shallow}}^{(\text{apical})}), \quad C_{\text{shallow}}^{(\text{apical})} \alpha_{\text{shallow}}^{(\text{apical})} / C_{\text{anterior}}^{(\text{apical})}, \quad (7.8)$$

$$\alpha_{\text{anterior}}^{(\text{basal})}, \quad \text{sign}(\alpha_{\text{shallow}}^{(\text{basal})}), \quad C_{\text{shallow}}^{(\text{basal})} \alpha_{\text{shallow}}^{(\text{basal})} / C_{\text{anterior}}^{(\text{basal})}, \quad (7.9)$$

$$C_{\text{anterior}}^{(\text{apical})} / C_{\text{anterior}}^{(\text{basal})}, \quad C_{\text{shallow}}^{(\text{apical})} \alpha_{\text{shallow}}^{(\text{apical})} / (C_{\text{shallow}}^{(\text{basal})} \alpha_{\text{shallow}}^{(\text{basal})}), \quad (7.10)$$

где (7.8) — характеристики апикальных профилей, (7.9) — характеристики базальных профилей, а характеристики (7.10) показывают отношения между апикальными и базальными профилями.

Здесь и далее мы используем следующие характеристики модели, основанные на этих комбинациях:

- показатель переднего градиента $\lambda_{\text{anterior}}^{(\text{apical})} = \exp(\alpha_{\text{anterior}}^{(\text{apical})})$ и $\lambda_{\text{anterior}}^{(\text{basal})} = \exp(\alpha_{\text{anterior}}^{(\text{basal})})$;
- логарифм отношения предэкспоненциальных коэффициентов переднего градиента для апикального и базального профилей $C^{\text{ab}} = \ln(C_{\text{anterior}}^{(\text{apical})} / C_{\text{anterior}}^{(\text{basal})})$;
- индикаторы неувеличения неглубоких компонентов $\lambda_{\text{shallow}}^{(\text{apical})} \leq 1$ и $\lambda_{\text{shallow}}^{(\text{basal})} \leq 1$.

Заметим, что эти характеристики имеют смысл, если $C_{\text{anterior}}^{(\text{apical})}$, $C_{\text{anterior}}^{(\text{basal})}$, $C_{\text{shallow}}^{(\text{apical})}$ и $C_{\text{shallow}}^{(\text{basal})}$ положительны. Страна модель, мы предположили, что передняя экспонента практически обнуляется в пределах длины эмбриона. Чтобы проверить это, рассмотрим дополнительные характеристики: положения AP $AP_0_{\text{anterior}}^{(\text{apical})}$ и $AP_0_{\text{anterior}}^{(\text{basal})}$, при которых передние экспоненты становятся почти нулевыми (точнее, такой процент $x \% \text{EL}$, что $C\lambda^x < 1$; напомним, что интенсивность измеряется в диапазоне 0–255). На эти характеристики может повлиять масштабирование, поэтому их следует использовать осторожно.

Эти соотношения лежат в основе количественных выводов в данной работе. Мы также используем эти соотношения для скрининга нетипичных эмбрионов и выбросов, что помо-

гает проследить развитие *Bcd* градиента РНК с течением времени, а также для изучения апикально-базальных различий.

Оценка параметров двухэкспоненциальной модели Мы используем метод ESPRIT на основе подпространств, мотивированный успехом SSA (тоже метода на основе подпространств) в сглаживании одномерных профилей генов из эмбрионов дрозофилы, см. [199] и раздел 7.2 с примерами, где SSA применяется для сглаживания. На профилях различных генов метод оказался устойчивым к высоким шумам и вариациям характеристик эмбрионов.

Математические детали ESPRIT можно найти в разделе 1.9.2. Мы используем метод для оценки экспоненциального спада в (7.6): $\lambda_{\text{anterior}}^{(\text{apical})}$, $\lambda_{\text{anterior}}^{(\text{basal})}$, $\lambda_{\text{shallow}}^{(\text{apical})}$ и $\lambda_{\text{shallow}}^{(\text{basal})}$. Оценка коэффициентов $C_{\text{anterior}}^{(\text{apical})}$, $C_{\text{anterior}}^{(\text{basal})}$, $C_{\text{shallow}}^{(\text{apical})}$ и $C_{\text{shallow}}^{(\text{basal})}$ затем выполняется по обычному методу наименьших квадратов, поскольку модель (7.6) линейна по предэкспоненциальным коэффициентам.

Так как первая экспонента должна быть быстро убывающей ($\lambda_{\text{anterior}}^{(\text{apical/basal})} < 1$), а второй экспоненциальный показатель, как ожидается, будет близок к постоянному ($\lambda_{\text{shallow}}^{(\text{apical/basal})}$ около 1), мы меняем порядок оценок ESPRIT двух экспоненциальных показателей соответствующим образом.

Данные

FISH и сбор данных Для подтверждения предложенной модели мы рассмотрели набор данных конфокальных изображений ранних эмбрионов дрозофилы, окрашенных флуоресцентом на мРНК *bcd* (база данных FISH). Эти данные были охарактеризованы в работе [200], и это самый большой подобный набор данных на сегодняшний день. Наш набор данных состоит из изображений около 160 эмбрионов, начиная от неоплодотворенных яиц (не анализировались) и заканчивая ранним циклом ядерного расщепления 14A (nc14, тот же набор данных, что и в [200]). В данном исследовании мы проанализировали 124 эмбриона. На основании предварительного анализа и биологических соображений они были разделены на три стадии развития: Cleavage, или предblastодерма (nc1–nc9); Синцитиальная бластодерма (nc10–nc13); и Клеточная бластодерма (nc14A). Стадия расщепления длится долго, около 80 минут (при комнатной температуре), и имеет сильно изменяющиеся градиенты *bcd* мРНК. Для более детального анализа мы разделили стадию расщепления на две подгруппы: Раннее (nc1–nc8) и Позднее (nc9). Стадия синцитиальной бластодермы длится около 45 мин, и ее можно разделить на две подгруппы: nc10–nc12 и nc13. Последняя стадия, ранняя nc14A, коротка (15–20 мин.), но очень изменчива и динамична. Тщательный визуальный осмотр позволил нам разделить эмбрионы nc14A на три подгруппы: ранние, средние и поздние [200].

Построение 1D профилей Необработанные данные с конфокального микроскопа состоят из интенсивности мРНК на небольшой круговой области с двумерными пространственны-

ми координатами. После выделения областей интереса (цепочек ROI) были опробованы две техники для преобразования данных в 1D профили AP. Первый (и самый простой) метод проецирует интенсивность на ось AP, ортогональную оси DV, отбрасывая компонент DV координаты (рис. 7.10A). Этот метод используется многими группами, см. например [203, 198]. Второй метод сохраняет естественные криволинейные координаты эмбриона, при этом расстояние между ROI рассчитывается по формуле $d^2(i) = (AP(i+1) - AP(i))^2 + (DV(i+1) - DV(i))^2$. Кумулятивные расстояния затем нормализуются путем деления на сумму $d(i)$.

Независимо от метода (AP или криволинейные координаты), полученные одномерные координаты не являются равноудаленными. Для создания равноудаленных точек с заданным пространственным шагом использовалась линейная интерполяция. Шаг 0,08–0,1%EL был выбран для создания примерно одинакового количества точек для двух методов. Результаты, полученные с помощью AP координат, кажутся более точными, чем результаты, полученные с помощью криволинейных координат. Поэтому здесь мы можем рассматривать только AP координаты.

7.3.2. Результаты и обсуждение

Применение модели

Рис. 7.11 демонстрирует набор примеров, иллюстрирующих разнообразие профилей, которым может соответствовать двухэкспоненциальная модель. Они включают типичные профили, рассмотренные в работе [200], с быстро уменьшающимся передним градиентом и медленно уменьшающимся градиентом к задней части (рис. 7.11A), и профили с увеличивающимся (рис. 7.11B) или плоским (рис. 7.11C) задним градиентом.

Данные из терминальных областей эмбриона: 0–10 %EL и 90–100 %EL, как правило, слишком смещены и зашумлены (ср. рис. [198] и рис. 7.11). Обработки и анализа данных с 10–90 %EL оказалось достаточно для извлечения *Bcd* профилей РНК почти всех эмбрионов старше nc6. Для очень ранних эмбрионов (стадия CleavageEarly), градиенты только начали формироваться из начальных терминальных локализаций; в этих случаях может быть более целесообразно обрабатывать данные с 5 %EL (Рисунок 7.11D). Для единобразия мы будем обрабатывать все данные на 10–90 %EL. Рис. 7.11 показывает, что двухэкспоненциальная модель очень хорошо подходит для различных типов данных.

Как правило, эмбрионы имеют уменьшающийся передний экспоненциальный компонент и уменьшающийся или близкий к постоянному неглубокий задний градиент (как для апикального, так и для базального профилей). Такие эмбрионы мы называем эмбрионами типа 1 (типичными). Некоторые эмбрионы, однако, демонстрируют увеличивающийся в заднем направлении неглубокий градиент как для апикального, так и для базального профилей. Мы называем такие эмбрионы эмбрионами типа 2 (атипичные). Профили типа 2 распространены на ранних стадиях развития (Cleavage) и редко встречаются на более поздних стадиях. Здесь мы сосредоточимся на эмбрионах типа 1, которые составляют большинство набора

данных.

Обнаружение типа 1 может быть осуществлено с помощью экспоненциальных показателей малых экспонент: (A) $\lambda_{\text{shallow}}^{(\text{apical})} < 1.002$, $\lambda_{\text{shallow}}^{(\text{basal})} < 1.002$. Это условие ищет неглубокие профили (как базальные, так и апикальные), которые не увеличиваются к задней части (1.002 используется вместо 1, чтобы учесть ошибки оценки).

Проверка модели

Даже в пределах одной стадии развития форма профилей мРНК от эмбриона к эмбриону очень изменчива. Это делает построение прототипа профиля сложной задачей и затрудняет понимание биологических механизмов, лежащих в его основе. К счастью, изменчивость в основном обусловлена меньшинством эмбрионов, и их можно обнаружить с помощью двухэкспоненциальной модели. Удаление таких эмбрионов значительно снижает изменчивость.

Мы удаляем эмбрионы, которые не были описаны моделью с разумными параметрами. Во-первых, предэкспоненциальные коэффициенты для передней и малой экспоненты должны быть положительными. Поэтому мы предполагаем, что (B) $C_{\text{anterior}}^{(\text{apical})} > 0$, $C_{\text{anterior}}^{(\text{basal})} > 0$, $C_{\text{shallow}}^{(\text{apical})} > 0$ и $C_{\text{shallow}}^{(\text{basal})} > 0$. Тогда для обычных эмбрионов передние экспоненты должны доходить до почти нуля в диапазоне 0–100 %EL. Поэтому мы накладываем условие (C) $AP0_{\text{anterior}}^{(\text{apical})} \leq 100$, $AP0_{\text{anterior}}^{(\text{basal})} \leq 100$. Обратим внимание, что (A), (B), (C) устойчивы к небольшим изменениям порогов ограничений (результаты не показаны).

92 эмбриона удовлетворяют условиям (A)–(C) из полного набора данных 124 эмбрионов; анализ в остальной части данного исследования проводится на этих 92 эмбрионах. Для этих 92 эмбрионов было проверено, что систематические ошибки в модели пренебрежимо малы по сравнению с остаточным шумом или самим профилем. Таким образом, мы пришли к выводу, что профили соответствуют рассматриваемой модели с достаточной точностью.

Эффективность модели для поиска тенденций в биологии развития

Параметры, полученные в результате двухэкспоненциальной подгонки, весьма изменчивы, как в пределах, так и между стадиями развития (Рис. 7.12А), как и ожидалось от наблюданной изменчивости профилей.

Хотя большая изменчивость и небольшой размер выборки не позволяют сделать статистически значимые выводы для всех сравнений, можно сделать несколько наблюдений. В частности, CleavageEarly имеет наибольшую среднюю переднюю константу экспоненциального спада среди всех стадий развития (т.е. самый крутой профиль). Это различие статистически значимо (t-тест), но может оказаться незначительным из-за умеренных изменений только у одного из шести эмбрионов. Поэтому мы объединили группы, чтобы получить 3 возрастные группы (из 7) с большим объемом выборки: (1) Cleavage, $n = 20$; (2) nc10–nc13, $n = 25$; (3) nc14, $n = 47$. Рис. 7.12В показывает, что эти большие группы имеют более четкую кластеризацию, с различными средними значениями.

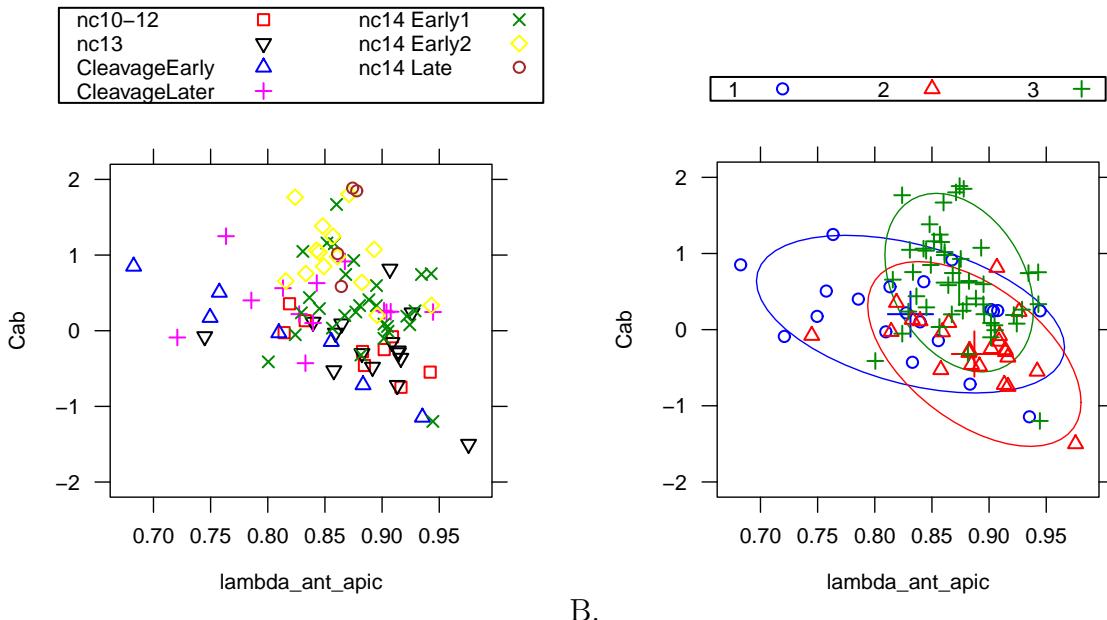


Рис. 7.12: Предэкспоненциальный множитель C^{ab} против переднего градиента $\lambda_{\text{anterior}}^{(\text{apical})}$. **A.** 7 (отмеченных) стадий развития (Раздел 7.3.1): Двухэкспоненциальные параметры показывают большую изменчивость между и внутри стадий развития. **B.** 3 объединенные группы (см. легенду): разница в значениях параметров, с 80% эллипсоидами рассеяния.

Таблица 7.7 показывает средние значения для $\lambda_{\text{anterior}}^{(\text{apical})}$ и C^{ab} с их 90% доверительными интервалами.

Таблица 7.7: Объединенные группы: средние значения и 90% доверительные интервалы для основных характеристик апикального и базального профилей для 3 групп.

	$\lambda_{\text{anterior}}^{\text{(apical)}}$	нижняя граница	верхняя граница
Cleavage	0.831	0.803	0.859
nc10-13	0.887	0.870	0.904
nc14	0.874	0.865	0.883
	C^{ab}	нижняя граница	верхняя граница
Cleavage	0.203	-0.010	0.417
nc10-13	-0.320	0.543	-0.096
nc14	0.618	0.461	0.775

Однофакторный дисперсионный анализ ANOVA (как параметрический, так и непараметрический (Kruskal-Wallis)) подтверждает, что $\lambda_{\text{anterior}}^{\text{(apical)}}$ и C^{ab} значимо отличаются между группами на уровне 5%. Post-hoc сравнения показывают, что C^{ab} (логарифм отношения между апикальным и базальным передними градиентами при 10 %EL) значимо отличается между всеми тремя группами; в то время как экспоненциальная скорость распада переднего градиента значимо больше (λ меньше) только для группы Cleavage.

Возможности подхода В разделе 7.3.2 мы отбирали эмбрионы в тип 1, используя условие (A), т.е. с неувеличивающимся в хвостовой части профилем. Мы можем применить предложенный подход к эмбрионам типа 2 с возрастающими на хвосте профилями (см. рис. 7.11B). Более того, модель может быть расширена до трех экспоненциальных профилей (см. рис. 7.13). С расширением SSA для соответствия трехэкспоненциальной модели, эти типы паттернов могут быть легко проанализированы текущим подходом, расширяя использование метода для сравнения паттернов различных генов (например, можно рассмотреть белок Stau [200], который имеет резкий подъем вблизи заднего полюса).

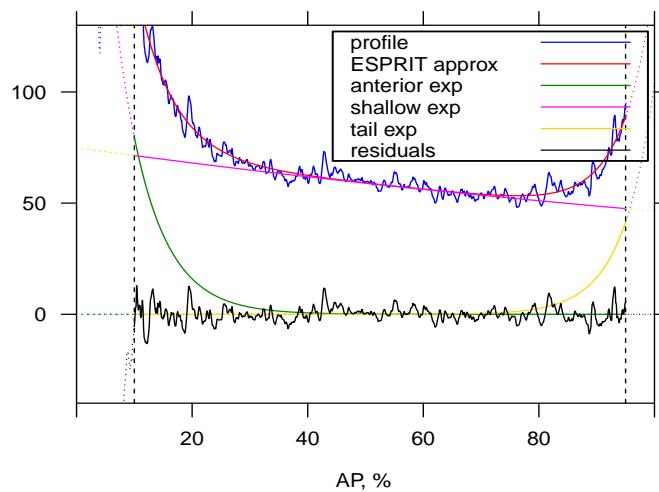


Рис. 7.13: АР-профиль белка Stau (ср. [200, Fig.6]). Исходные данные зашумлены; график получен с помощью 3-экспоненциальной модели: передняя экспонента 1, убывающая до нуля слева; мелкая экспонента 2, похожая на прямую линию; хвостовая экспонента 3, возрастающая от нуля справа; остаточный шум колеблется вокруг нуля

Представленный здесь подход, вероятно, станет эффективным инструментом для количественной оценки других пространственных градиентов в биологии развития, что может помочь в выявлении новых особенностей в динамике паттернинга и регуляции критических событий развития, особенно там, где наблюдаются большие динамические изменения и высокая изменчивость — т.е. в случаях, когда трудно построить эталонный или прототипный профиль.

7.4. Shaped SSA для количественной оценки экспрессии генов, с применением к раннему эмбриону дрозофилы

В последние годы, с развитием технологий автоматизированной микроскопии, объем и сложность данных изображений об экспрессии генов чрезвычайно возросли. Единственным

способом количественного и всестороннего анализа таких биологических данных является разработка и применение новых сложных математических подходов. Здесь мы представляем расширения двумерного сингулярного спектрального анализа (2D-SSA) для применения к двумерным и трехмерным наборам данных изображений эмбрионов. Эти расширения, Circular и Shaped 2D-SSA, применяются к экспрессии генов в ядерном слое непосредственно под поверхностью эмбриона *дрозофилы* (плодовая муха). При анализе мы будем рассматривать обычно используемую цилиндрическую проекцию эллипсоида эмбриона *дрозофилы*. Мы продемонстрируем, как Circular и Shaped версии 2D-SSA помогают разложить данные об экспрессии на идентифицируемые компоненты (такие как тренд и шум), а также разделить сигналы от разных генов. Рассмотрим вопросы обнаружения и улучшения недостаточной и избыточной коррекции при многоканальной визуализации, а также извлечения и анализа трехмерных особенностей в трехмерных графиках экспрессии генов.

В данной работе мы представляем новые вычислительные инструменты для анализа паттерна экспрессии генов для наборов данных трех пространственных измерений, примененных к ранним эмбрионам *дрозофилы*. Эти инструменты являются расширением двумерного сингулярного спектрального анализа (2D-SSA).

Сингулярный спектральный анализ выбран, поскольку он не требует априорного задания модели шума. Мы разлагаем ряд данных на множество элементарных рядов, анализируем их, выбираем подходящие компоненты и, наконец, суммируем идентифицируемые компоненты в классы. Например, выбор гладких компонентов может привести к адаптивному сглаживанию. SSA очень полезен для исследовательского анализа, поскольку метод может работать с модулированным шумом, т.е. шумом, который может зависеть от значений тренда (например, иметь мультиплексную природу).

В последнее время SSA был расширен для анализа двумерных объектов (2D-SSA), например, цифровых изображений [66, 40]. Разложение изображений является более сложной задачей по сравнению с анализом временных рядов из-за изменчивости 2D паттернов. Но методы, которые легко контролируются и адаптируются, такие как 2D-SSA, могут иметь широкое применение. 2D-SSA и связанные с ним методы на основе подпространств применяются в анализе текстур [178], сейсмологии [179], пространственных данных экспрессии генов [6], и медицинской визуализации [204]. В работе [43] 2D-SSA был применен для анализа цифровых рельефов в геологии и было продемонстрировано, что 2D-SSA является полезным инструментом для анализа различных уровней деталей в данных, измеренных на некоторой плоской поверхности. Позже 2D-SSA был применен к данным экспрессии генов, чтобы отделить ядерный шум от тенденции экспрессии [6].

В работах [12, 41] представлены расширения 2D-SSA, которые расширяют спектр применения SSA. В данном разделе мы демонстрируем, как Shaped 2D-SSA может быть применен для анализа данных экспрессии генов [10]. Описанные здесь новые подходы, Circular и Shaped 2D-SSA, особенно подходят к цилиндрическим поверхностям (как это используется для эмбрионов *дрозофилы*), чтобы избежать краевых эффектов, и к области данным непрямо-

угольной формы. Например, область данных хорошего качества на изображении (например, без перенасыщения) может быть не прямоугольной и даже иметь пропуски. Кроме того, поскольку плоскостная проекция эмбриона дрозофилы почти эллиптическая, способность анализировать непрямоугольные формы может быть полезной.

7.4.1. Материалы

Данные взяты из проекта Berkeley Drosophila Transcription Network Project (BDTNP) [205], который содержит трехмерные (3D) измерения относительной концентрации мРНК для 95 генов в раннем развитии (включая *snail* (*sna*)), а также схемы экспрессии белка для четырех генов (*bicoid*, *giant*, *hunchback* (*hb*) и *Krüppel* (*Kr*)) во время ядерных циклов расщепления 13 (C13) и 14 (C14A). BDTNP Release 2 содержит индивидуальные наборы данных (файлы PointCloud) для 2830 эмбрионов (<http://bdtnp.lbl.gov/Fly-Net/bioimaging.jsp>). Эти данные были измерены в координатах 6078 ядер на коре эмбриона и представлены в виде интегрированного набора данных (файл VirtualEmbryo, с инструментами для визуализации и анализа). Эмбрионы фиксировали и флуоресцентно окрашивали для маркировки экспрессии мРНК двух генов и ДНК в ядрах. Один из окрашенных генов был либо *even skipped* (*eve*), либо *fushi tarazu* (*ftz*), которые использовались как фидуциарные маркеры для последующей пространственной регистрации.

7.4.2. Выбор параметров, разделимость и идентификация компонентов

Подход к выбору размера окна для одномерных временных рядов подробно описан в [1], см. также раздел 3.4. Рекомендации для двумерных объектов сложнее. Для извлечения так называемых объектов конечного ранга (суммы произведений многочленов, экспонент и синусоид), которые удовлетворяют линейным рекуррентным соотношениям (ЛРС), окна должны быть большими, вплоть до половины размера объекта. Однако реальные детали обычно имеют сложную форму и удовлетворяют ЛРС только приближенно и локально. Окно должно согласовываться с этим локальным характером. В частности, синусоиды точно подчиняются ЛРС. Однако если двумерная синусоида имеет медленно меняющееся местоположение, то только ее локальные части удовлетворяют ЛРС. Размеры окна должны соответствовать масштабу этой локальности. Выбор размера окна — это всегда баланс между локальным и глобальным масштабом данных.

В целом, SSA может отделить гладкие паттерны от шума для широкого спектра паттернов. Для регулярных паттернов 2D-SSA может применяться независимо от того, плавно или резко изменяется паттерн. Однако если паттерн нерегулярный, вариации должны быть плавными, чтобы использовать 2D-SSA для разделения сигналов. Нерегулярный паттерн с резкими изменениями плохо выделяется 2D-SSA. Если, однако, резкое изменение происходит в узкой области, ее можно вырезать, а оставшиеся данные проанализировать с помощью Shaped SSA, которая представляет собой версию 2D-SSA с непрямоугольной формой изобра-

жения или окна.

Элементарные компоненты группируются на основе их сходства с извлекаемыми компонентами данных. Для регулярных компонентов, таких как синусоиды, количество элементарных компонентов может быть рассчитано из теории. Кроме того, регулярные компоненты обычно имеют ограниченный частотный диапазон (обычно в нем отсутствуют высокие частоты). Поэтому в общем случае паттерну приписываются ведущие элементарные компоненты с соответствующими частотными характеристиками.

В данном исследовании мы показываем, как 2D-SSA можно использовать для удаления шума, отделения регулярных колебаний от медленно меняющихся паттернов (для коррекции ошибочных выполненные процедуры удаления смешивания сигналов) и выделения полос для их дальнейшего анализа. Shaped SSA позволяет анализировать сложные паттерны путем разделения изображений на несколько частей.

Ранняя экспрессия генов *дрозофилы* (до перехода в среднюю бластулу) создает гладкие и простые паттерны, подходящие для обработки 2D-SSA. На ряде веб-ресурсов есть такие наборы данных (BDTNP BID, Fly-FISH, FlyEx и другие). Shaped SSA также может быть полезен для подмножества этих данных, в котором паттерны резко падают до нуля. В этих случаях субрегионы могут быть вырезаны или проанализированы отдельно от всего изображения. Ген *sna* является типичным примером гена *дрозофилы*, наблюдаемым в BDTNP BID; такие компактные паттерны также наблюдаются в других экспериментальных организмах, например, девять генов зебрафиш [206]. Поэтому мы ожидаем, что 2D-SSA и Shaped SSA будут широко применяться для обработки изображений в биологии развития.

Проблема смешивания паттернов экспрессии двух разных генов на одном изображении [207] требует дополнительных условий. В частности, необходима информация о несмешанной экспрессии каждого гена (т.е. данных одного гена в отсутствие другого гена). Если два гена имеют медленно меняющиеся паттерны, они не могут быть легко разделены с помощью SSA. В таких случаях SSA нельзя использовать для обнаружения или исправления ошибок в смешанных изображениях. Однако SSA является эффективным методом разделения смешивания для случаев, когда один ген имеет близкую к регулярной структуру, которая отличается от структуры другого гена. В данной работе мы применяем SSA для разделения сигналов и коррекции изображений для таких случаев на основе данных *дрозофилы*.

Предобработка данных

Первоначально данные для 2D-SSA анализа должны быть измерены на регулярной сетке. Данные по экспрессии генов измеряются в ядрах, которые не расположены регулярно на трехмерной поверхности эмбриона (которая имеет близкую к эллипсоидальной форму). Первым шагом предварительной обработки является цилиндрическая проекция данных (с центром на главной оси эллипса; главная ось эмбриона находится с помощью анализа главных компонент). Затем мы интерполируем данные на регулярную сетку на этом цилин-

дре. Мы анализируем центральную область цилиндра, чтобы избежать повреждений вблизи полюсов в результате преобразования эллипсоида в цилиндр. После 2D-SSA разложения мы интерполировали данные обратно на ядерные центры. Эта интерполяция выполняется для гладких компонентов; остатки рассчитываются как разница между исходными данными и интерполированными гладкими компонентами.

Интерполяция включает триангуляцию Делоне с последующей линейной интерполяцией ядерных центров в триангуляцию.

Реализация

Алгоритмы реализованы в пакетах RSSA и BioSSA в R. Алгоритмы 2D-SSA требуют больших затрат времени и памяти, поэтому очень важно иметь эффективную реализацию. R-пакет BioSSA является дополнением к RSSA для применения к данным экспрессии генов эмбрионов мух и кратко описан в <http://biossa.github.io/>.

7.4.3. Остаточные периодические паттерны, создаваемые алгоритмами разделения сигналов

Различные спектры излучения для флуоресцентных зондов позволяют одновременно окрашивать 3–4 генных продукта в эмбриональных тканях. Проекты количественной визуализации [203, 205] используют один и тот же ген в одном из этих каналов во всех эмбрионах, для надежных количественных сравнений, регистрации и т.д. Ген, используемый для такой маркировки у эмбрионов *дрозофилы*, обычно является одним из pair-rule генов (таких как *eve* или *ftz*), которые имеют характерный периодический 7-полосный паттерн экспрессии.

Многоканальная визуализация страдает от неустранимой проблемы наложения спектров эмиссии (когда флуоресцентные маркеры возбуждаются одновременно), когда свет от более чем одного флуоресцентного красителя собирается одним каналом регистрации. Для уменьшения перекрестных помех с помощью вычислений, был разработан и применен к данным BDTNP метод автоматического разделения каналов.

Проблема этого подхода в крупномасштабных проектах с автоматической обработкой данных заключается в том, что оцененные параметры смешивания могут оказаться слишком высокими или слишком низкими. Если параметры завышены, борьба со смешиванием приводит к избыточной коррекции, которая проявляется как частичное вычитание общего, эталонного образца из образца второго гена (исследуемого гена эмбриона). При периодических эталонных паттернах (*eve*, *ftz*), это приводит к образованию периодических борозд в “несмешанном” паттерне. На рис. 7.14 показан эффект такой перекоррекции в одном из эмбрионов BDTNP.

С другой стороны, если параметры разделения сигналов недооценены, то смешивание дает недокоррекция, которую можно рассматривать как добавление общего, эталонного, паттерна к паттерну второго гена (изучаемого в данном эмбрионе). На рис. 7.15 показан пример

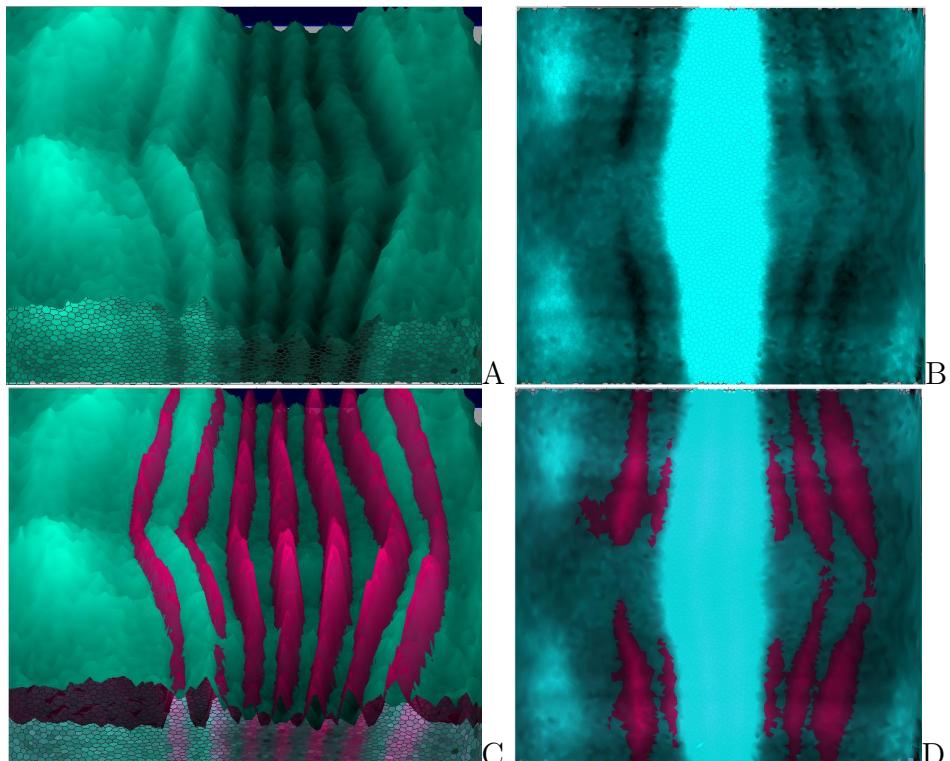


Рис. 7.14: Пример чрезмерной коррекции в данных экспрессии генов, приводящей к вычитанию эталонного генного паттерна (семиполосные паттерны *ftz* и *eve*; темно-пурпурный) от исследуемого паттерна (продукты генов *hb* и *Kr* (транскрипционные факторы); светло-голубой). Визуализация с помощью инструментов PointCloudXplore, эмбрионы BDTNP *hb* ‘v5-s11512-2oc06-25’ (А и С), *Kr* ‘v5-s12169-24oc07-22’ (В и Д); (С) — то же, что (А), с добавлением *ftz*; (Д) — то же, что (В), с добавлением *eve*

недокоррекции на эмбрионе BDTNP.

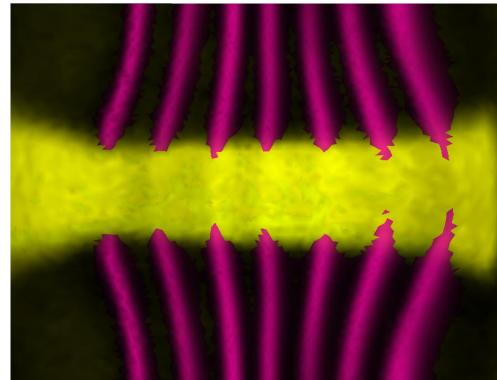


Рис. 7.15: Пример недостаточной коррекции, при которой периодический эталонный генетический паттерн (*eve*; темно-пурпурный) добавляет периодичность к непериодическому паттерну исследуемого образца (*sna* генетический продукт; желтый). Визуализация с помощью PointCloudXplore. Embryo ‘v5-s10531-28fe05-07’.

Видно, что неправильная оценка параметров смешивания вносит периодичность в ряде эмбрионов BDTNP из 7-полосных *eve* или *ftz* эталонных образцов. Этот эффект достаточно силен, чтобы быть заметным на некоторых изображениях, интегрированных из нескольких эмбрионов (например, рис. 7.15).

Теперь мы покажем, как разложение по Circular 2D-SSA может быть использовано для оценки и устранения периодических компонентов, вызванных недостаточной или избыточной коррекцией, на примере изображений BDTNP на рис. 7.14 и 7.15.

Circular 2D-SSA, *hb* испорченный *ftz*, сильная перекоррекция

На рис. 7.16 показаны оригинальные изображения для экспрессии генов *hb* и *ftz* для BDTNP эмбриона (ID ‘v5-s11512-2oc06-25’). Естественная тенденция *hb* — низкочастотная; естественный паттерн *ftz* — высокочастотный; перекрестные связи, с избыточной коррекцией в алгоритме смешивания, “вкрашивают” высокочастотный паттерн *ftz* в паттерн *hb*. Эти изображения “развернуты” из цилиндрической проекции данных; поэтому верхний и нижний края соединяются (периодические граничные условия).

Мы предварительно обрабатываем изображения, интерполируя их к регулярной сетке (шаг 0.5%); и удаляя 20% слева и 5% справа (чтобы сосредоточиться на области полос). Использование Circular 2D-SSA позволяет нам анализировать цилиндрический набор данных. Мы используем прямоугольное окно размером 25×10 . Учитывая регулярные колебания вдоль передне-задней (AP, горизонтальной) координаты, первый размер окна, 25, больше второго, 10.

На рис. 7.17 представлено 2D-SSA разложение на элементарные компоненты изображения для *hb*; на рис. 7.18 — для *ftz* (мы изобразили 26 самых больших компонентов; меньшие компоненты не были признаны значимыми для восстановления изображения). Рис. 7.17 содержит ряд компонентов с вертикальными полосами, вызванными или находящимися под

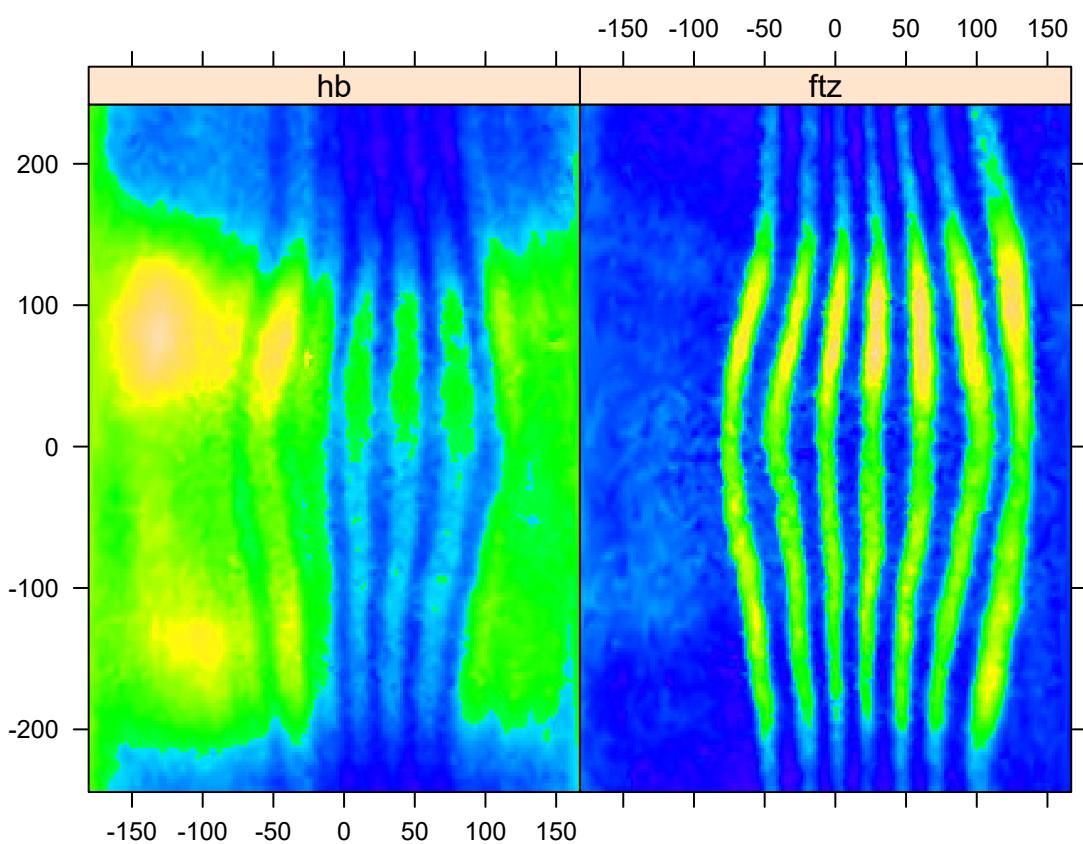


Рис. 7.16: hb и ftz : оригинальные изображения “развернутой” цилиндрической поверхности; верхние значения являются прямым продолжением нижних значений.

влиянием канала ftz . Если сравнить элементарные компоненты ftz -разложения (рис. 7.18, компоненты 2–5, 9–11, 15–17) с полосами с hb -разложением (рис. 7.17), то окажется, что hb -компоненты 1–4, вероятно, обусловлены паттерном экспрессии, тогда как компоненты 5–9, 11 и, возможно, 10, 12 обусловлены ftz -коррекцией.

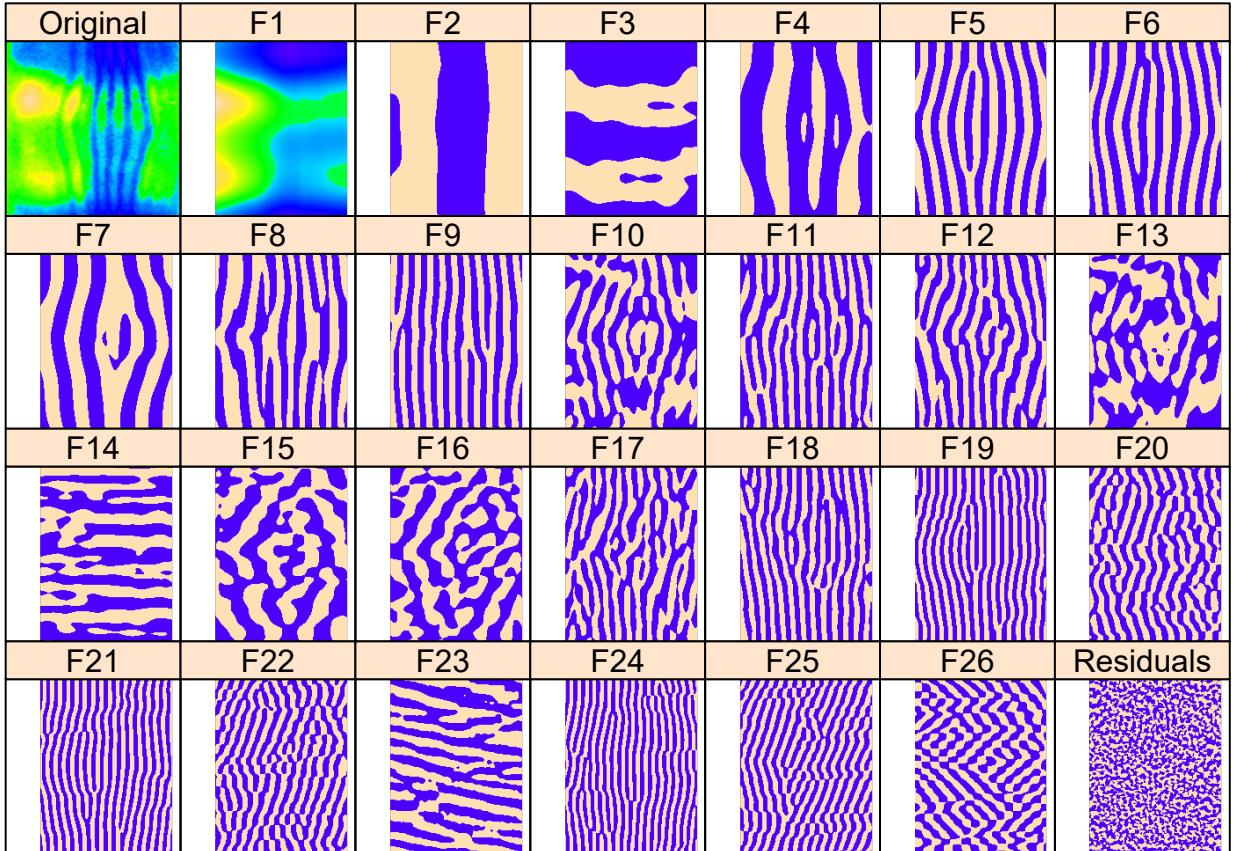


Рис. 7.17: hb : исходное изображение и элементарные компоненты, извлеченные Circular 2D-SSA. Исходное изображение и ведущий компонент (F1) выделены цветом в соответствии с минимальным и максимальным уровнями экспрессии. Для большей контрастности остальные компоненты изображены в бинарном формате: положительные значения — бежевым, отрицательные — фиолетовым.

На рис. 7.19 показано восстановление из ведущих высокочастотных компонентов для каждого изображения, компоненты 5 и 6 из рис. 7.17, компоненты 2 и 3 из рис. 7.18. Восстановления очень похожи, но имеют противоположные фазы, что указывает на то, что данные hb были чрезмерно скорректированы. Рис. 7.20 восстановлен из всех компонентов с полосами для каждого изображения; опять же, графики очень похожи, но имеют противоположные фазы.

Одновременно с удалением полос, этот процесс также разлагает изображение на паттерн и шум (остатки): На рис. 7.21 показано восстановление экспрессии гена hb из компонентов 1–4 без полос, наряду с компонентами с полосами (сильно затронутыми ftz) 5–12, и

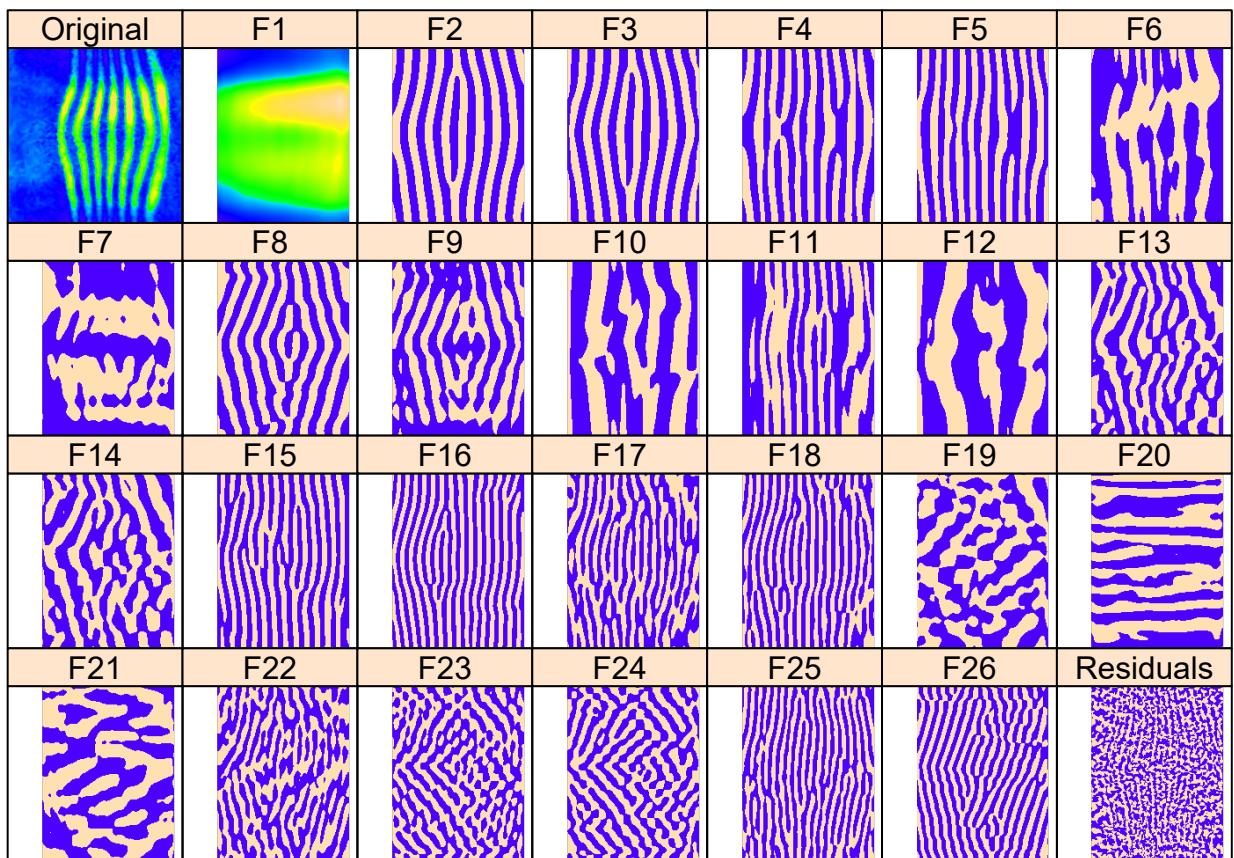


Рис. 7.18: ftz : исходное изображение и F1 с фоном; остальные элементарные компоненты изображены в бинарном формате.

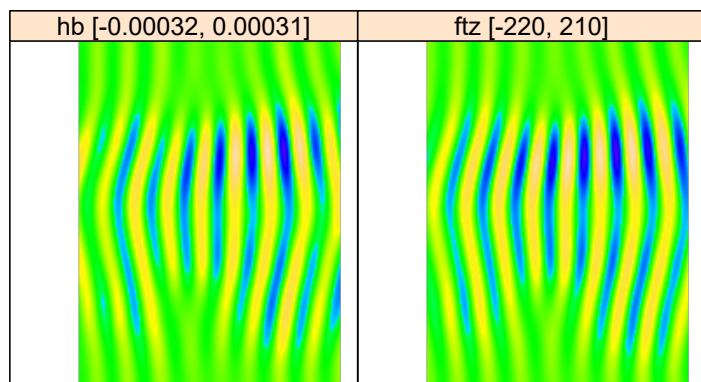


Рис. 7.19: hb (слева) и ftz (справа): восстановление из главных компонентов 5 и 6 с полосами для анализа hb , 2 и 3 для анализа ftz . Полосы находятся вне фазы для hb и ftz .

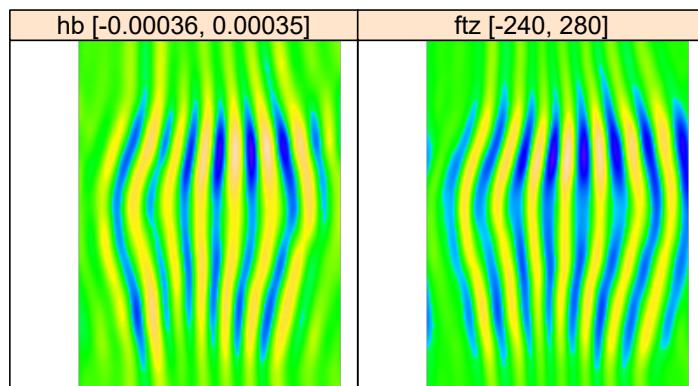


Рис. 7.20: hb (слева) и ftz (справа): восстановление из всех компонентов с полосами.

остатками. Circular 2D-SSA обеспечивает метод устранения недостаточной или избыточной коррекции в алгоритме смешивания и, следовательно, очистки генных паттернов от эффектов перекрестных связей. Для изображения без полос 2D-SSA дает прямое разложение на паттерн и шум. Здесь мы показали, что разложение SSA является работать устойчиво для данных с перекрестными полосами.

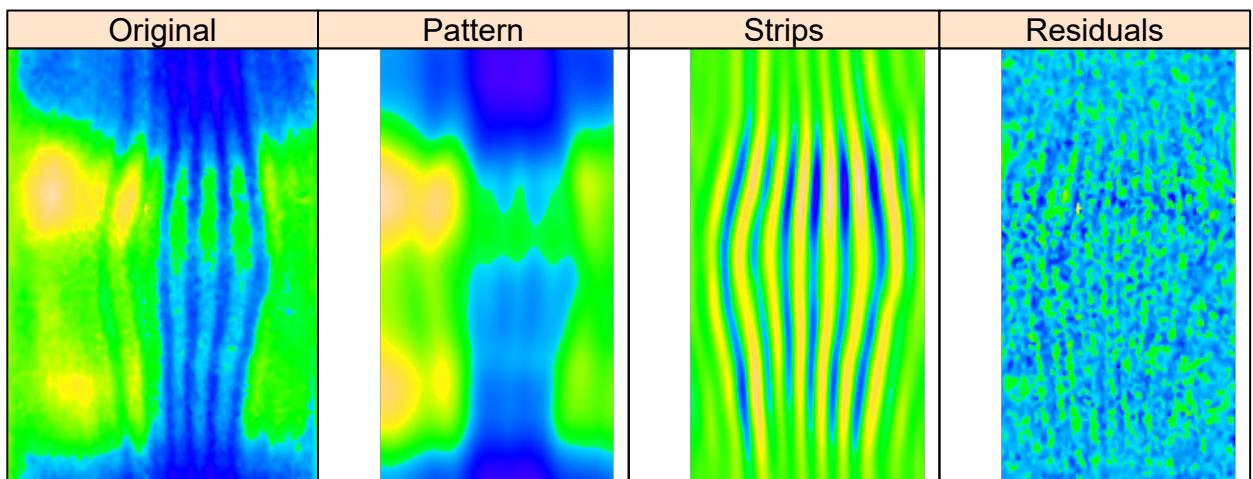


Рис. 7.21: hb , слева направо: исходное изображение, паттерн без полос, полосы и остаточный шум.

Shaped 2D-SSA, *sna* испорчена *eve*, недокоррекция

Ряд генов экспрессируется в паттернах, более сложных, чем вариабельность по оси AP, наблюдаемая для гар-генов, таких как hb и Kr . Для анализа перекрестных связей для таких данных мы вводим версию 2D-SSA для непрямоугольных изображений. В качестве примера, *snail* (*sna*) экспрессируется широкой полосой вдоль центральной линии эмбриона

(рис. 7.24, v5-s10531-28fe05-07, cy3_apical)). Поскольку *sna* показывает очень резкий переход от экспрессирующих к неэкспрессирующим областям, мы проанализировали их отдельно (рис. 7.22, экспрессирующие; рис. 7.23, неэкспрессирующие). Анализ проводился на регулярной сетке (шаг 0,5%), обрезанной по 15% слева и справа. Для центральной экспрессирующей зоны (рис. 7.22) мы использовали окно размером 40×10 ; для боковой неэкспрессирующей зоны (рис. 7.23) мы использовали окно размером 30×10 .

Разложение показывает, что элементарные компоненты {3,4} (рис. 7.22) и {4,5,16,17} (рис. 7.23) соответствуют полосам, которые происходят от референтного маркера *eve*. На рис. 7.22 и 7.23 показаны компоненты с полосами, а также эффект удаления этих полос для выявления сигнала *sna*. Рис. 7.24 показывает это для полного изображения *sna* (комбинация экспрессирующих и неэкспрессирующих зон). В этом случае компоненты полос с изображения *sna* и с изображения маркера *eve* находятся в фазе, что указывает на недокоррекцию в алгоритме разделения сигналов (см. рис. 7.25, где исходные изображения и восстановления полос собраны вместе).

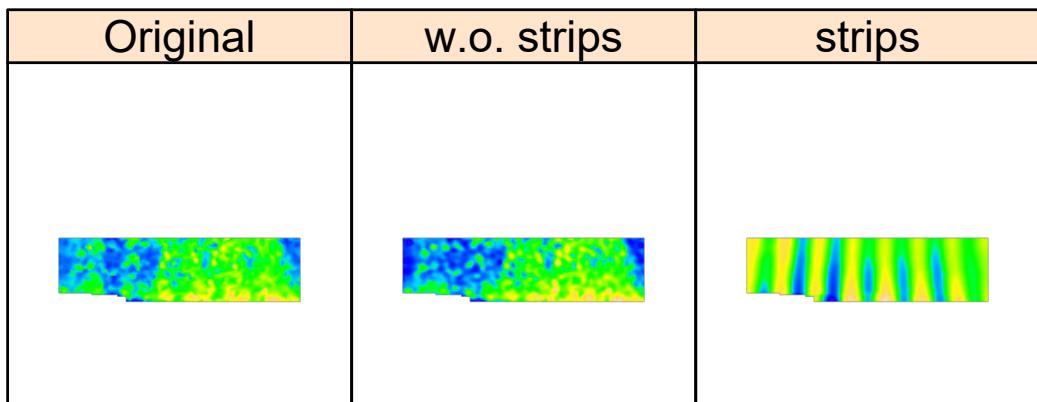


Рис. 7.22: *sna* изображение, область 1, зона сильной экспрессии. Слева направо: исходное изображение, восстановление без полос, и компоненты полос от маркера *eve*.

Таким образом, мы построили процедуру для удаления недо- или перекоррекции. Заметим, что если изображение не содержит полос, то изображения элементарных компонентов также не будут содержать полос и поэтому можно увидеть, необходима ли коррекция.

7.4.4. Выводы

Данное исследование показало применимость Shaped и Circular расширений 2D-SSA для анализа изображений эмбрионов из количественного высокопроизводительного проекта в биологии развития. Мы показали, что 2D-SSA может разлагать изображения и идентифицировать компоненты в соответствии с интересующим геном. Это эффективное средство для уменьшения “перекрестных помех” между генными каналами, которые возникают в техни-

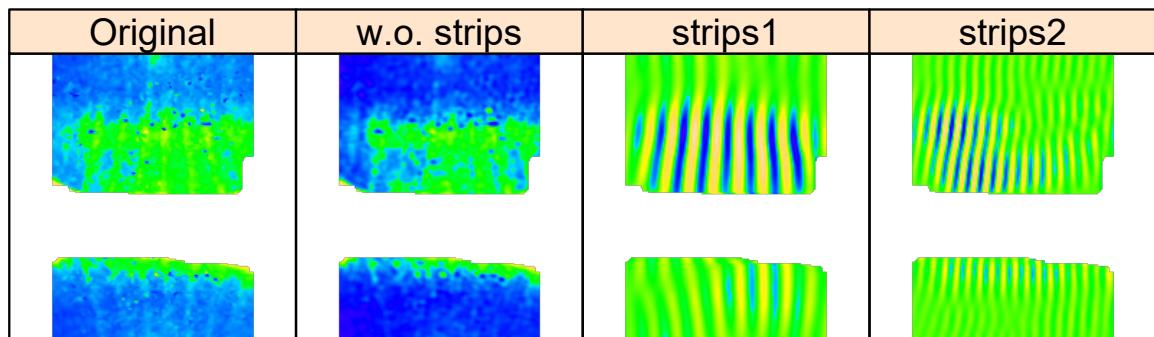


Рис. 7.23: *sna* изображение, область 2, зона слабой экспрессии. Слева направо: исходное изображение, восстановление без полос и компоненты полос.

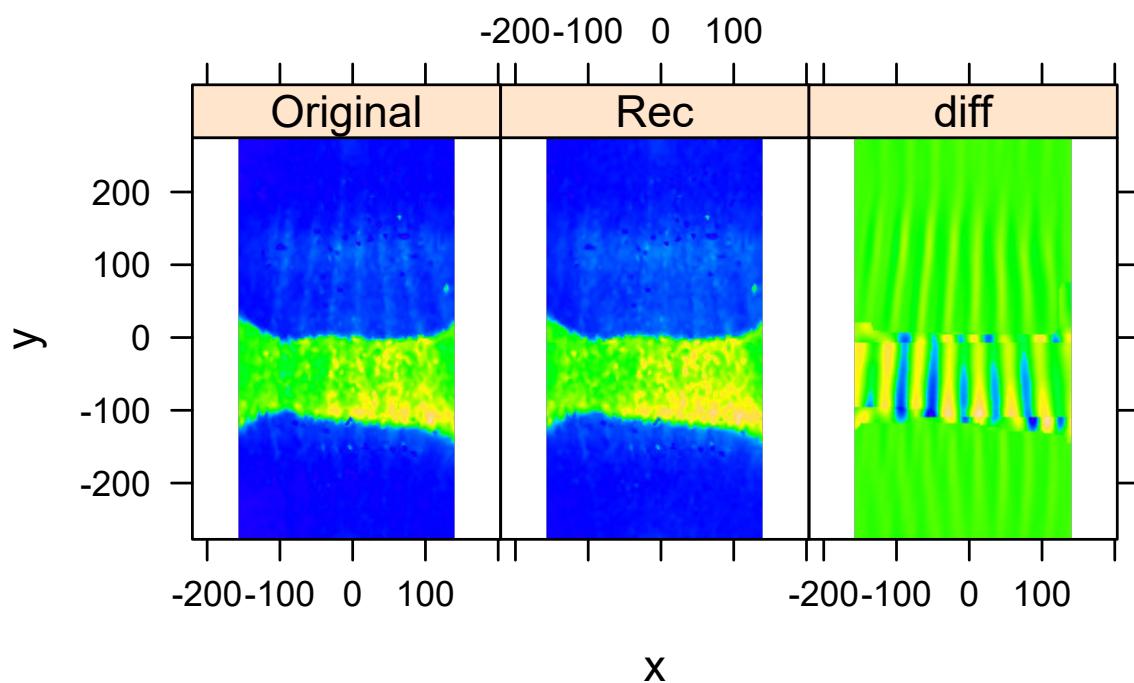


Рис. 7.24: *sna*, комбинированное изображение (обе зоны с рис. 7.22 и 7.23). Слева направо: исходное изображение, восстановление без полос и разница. Эмбрион BDTNP v5-s10531-28fe05-07.pce

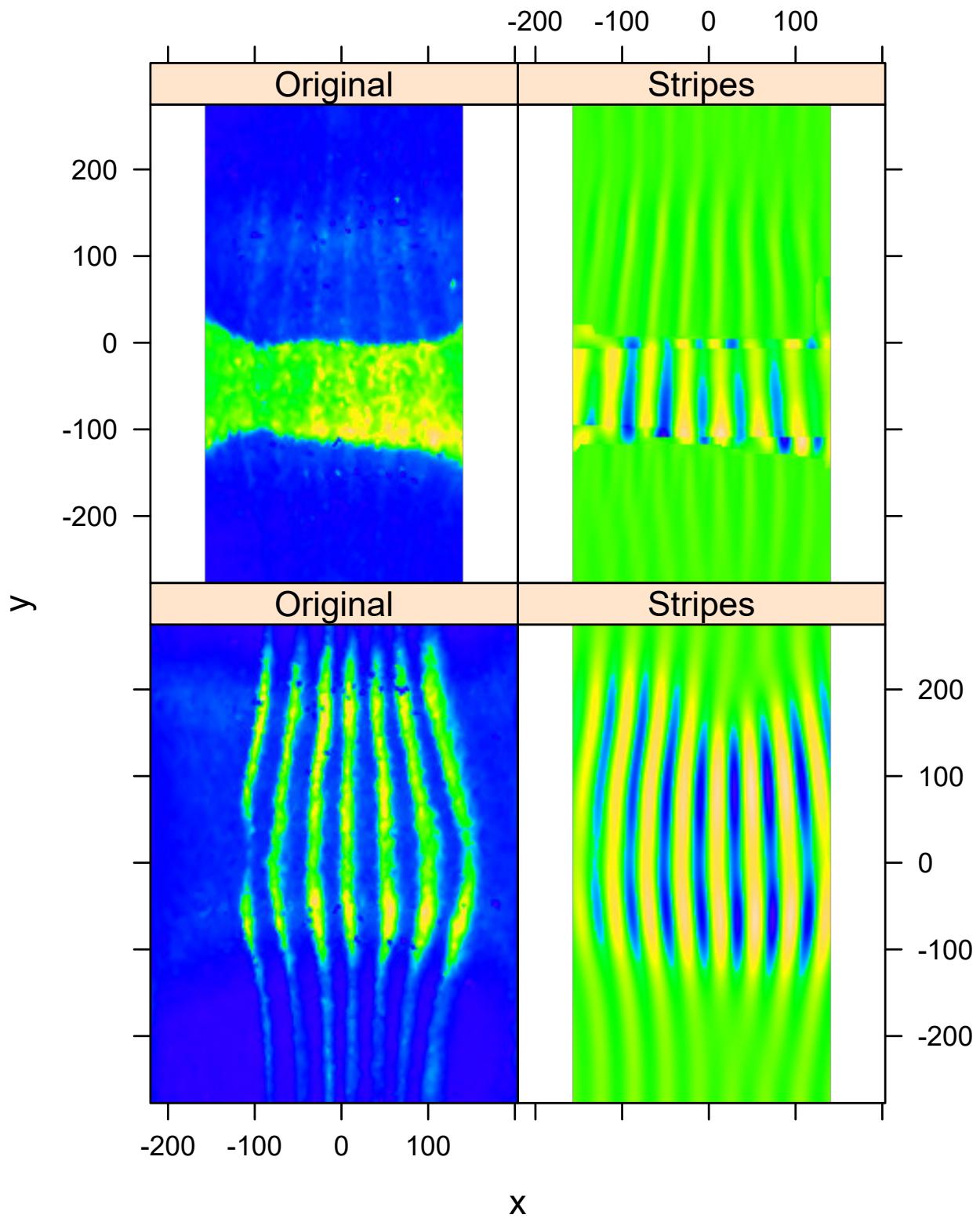


Рис. 7.25: *sna* и *eve*: оригинальные изображения (слева) и полоски (справа), *sna* сверху и *eve* снизу.

ке визуализации, но могут быть усилены автоматическим алгоритмом разделения сигналов после обработки.

Circular 2D-SSA является важным расширением для анализа цилиндрических данных проекций (рассматривая периодические границы в “прямоугольных” изображениях). Shaped 2D-SSA позволяет анализировать суб-регионы изображения, что важно для анализа сложных паттернов экспрессии, сложных геометрических форм данных и избегания краевых эффектов.

Процедура выполняется под контролем пользователя и может быть адаптирована к особенной структуре изображения с помощью гибкого выбора формы и размеров окна.

Мы продемонстрировали, что 2D-SSA может быть использован для извлечения сигнала и шума из изображений с сильной и слабой коррекцией перекрестных помех. Это является важным инструментом для разделения экспрессии генов в многоканальных изображениях, и для извлечения остаточного шума для изучения стохастических аспектов экспрессии генов. В частности, мы использовали SSA для разделения низкочастотных генов, представляющих интерес (gap-гены *hb* и *Kr*, а также *sna*) от “проникающего” перекрестного сигнала высокочастотных pair-rule фидуциарных маркеров (*eve* и *ftz*). Кроме того, мы показали, как компоненты SSA могут быть использованы для количественной оценки полос *eve* (в частности, полосы 4) для выявления новых типов изменчивости экспрессии, что приводит к новому пониманию механизмов развития. Это все примеры того, как может применяться 2D-SSA — мы ожидаем, что они будут широко обобщающимися для других случаев многоканальных 3D данных из *дрозофилы* и других организмов.

7.5. Shaped 3D SSA для количественной оценки экспрессии генов, с применением к раннему эмбриону зебрафиш

В этом разделе мы остановимся на анализе трехмерных объемных данных. Кроме того, что данные зашумлены, анализ осложняется тем, что клетки/ядра, могут быть неравномерно расположены в трехмерном пространстве. Это делает обработку, извлечение и изучение сигналов экспрессии и присущего им биологического шума серьезной проблемой для случая трехмерных данных, требующей новых вычислительных подходов. Здесь мы представляем новый подход для изучения экспрессии генов в ядрах, расположенных в толстом слое вокруг сферической поверхности. Метод включает выравнивание глубины на сфере, спрямление, интерполяцию на регулярную сетку, выделение паттерна с помощью Shaped 3D-SSA, и интерполяцию обратно к исходным позициям ядер. Подход продемонстрирован на нескольких примерах экспрессии генов в яйце зебрафиш (модельная система развития позвоночных). Метод протестирован на нескольких различных геометриях данных (например, позициях ядер) и различных формах паттернов экспрессии генов. Полностью трехмерные наборы данных для экспрессии генов в процессе развития становятся все более доступными; мы обсуждаем перспективы применения 3D-SSA для обработки и анализа данных в этой развивающейся

области.

Для восстановления пространственно-временной динамики экспрессии генов исходные наборы данных представляют собой последовательные в пространстве снимки ранних эмбрионов (фиксированных или живых), полученные с помощью конфокального микроскопа. Данные обычно представляют собой интенсивность флуоресцентных маркеров для мРНК или белков, кодируемых интересующими генами. Извлечение этих данных требует сегментации изображения для определения сигнала для каждой клетки (или ядра). В результате получаются текстовые файлы с пространственными координатами, уровнями экспрессии генов и историей развития каждой клетки.

Основной целью такой обработки является сбор надежных количественных данных для подгонки и проверки современных компьютерных динамических и стохастических моделей регуляции генов развития при одноклеточном разрешении.

Данные в трехмерных атласах экспрессии генов с клеточным разрешением обычно имеют очень высокий уровень шума, в который вносят вклад такие аспекты, как собственный шум экспрессии генов, наблюдаемый у прокариот и эукариот, и беспорядок в позициях клеток/ядер. Необходимы новые количественные подходы для разделения необработанных данных экспрессии на сигнальные и шумовые компоненты. В то время как эмбрионы некоторых животных имеют более простую геометрию, будучи относительно плоскими со сферическими или эллипсоидными клеточными слоями (как ранний эмбрион мухи дрозофилы), многие виды эмбрионов имеют изначально пространственно трехмерный порядок клеток, что добавляет методологические трудности из-за толщины образца и оптической непрозрачности. Такие трехмерные проблемы требуют новых экспериментальных и вычислительных подходов.

Различные геометрии эмбрионов могут создавать различные пространственные характеристики на данных экспрессии генов. Например, в раннем эмбрионе дрозофилы данные можно рассматривать как паттерны на эллипсоидной поверхности. При соответствующей двумерной развертке (например, цилиндрической проекции) данные могут быть изучены с помощью методов обработки двумерных изображений, см., например, раздел 7.4 и [208].

Однако существуют наборы последних экспериментальных данных, которые действительно являются трехмерными и не могут быть должным образом преобразованы и проанализированы в 2D. Ярким примером являются данные экспрессии, полученные на ранних стадиях развития эмбрионов зебрафиш, где ядра (клетки) находятся в нескольких неравномерных слоях на яйце рыбы. Такие данные требуют методик и алгоритмов для непосредственной обработки трехмерных данных. Неравномерное распределение ядер по слоям представляет собой дополнительную проблему, поскольку большинство количественных методов оперируют пространственно регулярными точками данных.

С этой целью в данной работе представлен новый метод обработки нерегулярных точек данных, рассеянных в слоях вблизи эллипсоидальной (сфероидальной) поверхности [11]. Мы представляем непараметрический метод, который может иметь дело с произвольным про-

странным распределением и неизвестным характером шума в данных об экспрессии. В разделе 7.4 мы представили расширения метода 2D-SSA для анализа 2D и поверхностных 3D наборов данных из конфокальных сканов *Drosophila*. Эти расширения, Circular и Shaped 2D-SSA, были применены для анализа экспрессии генов в тонком слое ядер под поверхностью эмбриона. Мы продемонстрировали, как Circular и Shaped 2D-SSA могут разложить данные экспрессии на идентифицируемые компоненты (тренд и шум), а также разделить сигналы от разных генов.

В данной работе мы расширяем SSA на нерегулярные трехмерные данные экспрессии (3D-SSA). Для первоначального применения подхода мы сосредоточились на работе с пространственными нерегулярностями — используя реальные данные зебрафиш для пространственных координат ядер и пространственных паттернов экспрессии генов из пакетов ‘MatchIT’ и ‘AtlasIT’ [206], но используя искусственные функции (простые математические функции и гладкую аппроксимацию показателя интенсивности) для интенсивности экспрессии (флуоресценции).

Рис. 7.26 и 7.27 иллюстрируют основные шаги подхода с применением 3D-SSA. (см. подробности в разделе 7.5.2). На рис. 7.26а показаны смоделированные данные интенсивности (двухэкспоненциальный паттерн, с шумом) на экспериментальных данных для позиций ядер в области экспрессии генов в геометрии сферической ‘шапочки’. На рис. 7.27 показаны шаги алгоритма: выравнивание глубины на сфере, спрямление, интерполяция, восстановление. На рис. 7.26б показано результатирующее восстановление паттерна после применения алгоритма 3D-SSA. Окраска ядер соответствует интенсивности экспрессии генов. Смешение ядер разных цветов отражает шум. Регулярные (гладкие) цветовые паттерны показывают результат удаления шума.

Это раздел структурирован следующим образом. Раздел 7.5.1 описывает полуискусственные наборы данных, которые были проанализированы. Подход с применением 3D-SSA описан в разделе 7.5.2. В разделе 7.5.3 демонстрируется надежность подхода на полуискусственных данных, похожих на реальные наблюдения.

В работе [11] было рассмотрено несколько примеров. В первом примере рассматриваются все ядра, обнаруженные в образце на стадии щитка, в котором ядра распределены в ‘сферической шапочке’, и экспрессия генов (с шумом) генерируется для двух паттернов: а) сумма двух экспонент и б) колоколообразная форма. Второй пример — для паттернов экспрессии, аналогичных паттернам для девяти регуляторных генов, охарактеризованных в [206]. В частности, тестовый паттерн ограничен ядрами, в которых ген *ntla* найден экспериментально (пакет ‘MatchIT’ [206]). Этот набор ядер распределен в экваториальной полосе. Для его выделения мы строим оболочку (огибающую) экспрессирующих ядер (обычно не выпуклую, так как область экспрессии имеет сложную форму). Shaped 3D-SSA применяется во всех тестовых случаях. В диссертации мы подробно опишем общую методику и первый пример из [11] с двух-экспоненциальным паттерном, который был рассмотрен для одномерного случая в [45].

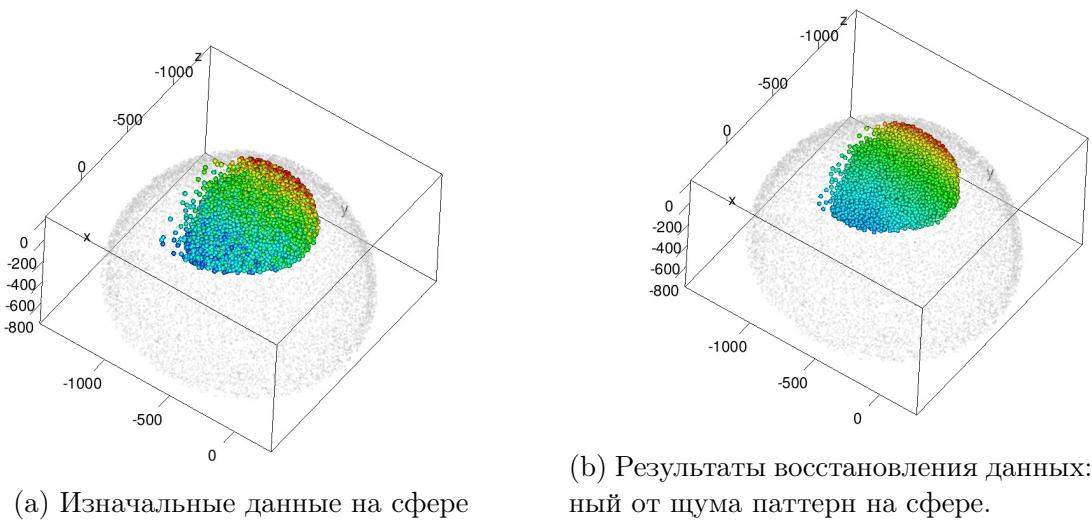


Рис. 7.26: Данные о зебрафиш и подход 3D-SSA: исходные данные о положении ядер в области экспрессии генов со сферической крышкой, с модельными значениями интенсивности (а схематичный двухэкспоненциальный паттерн, с шумом), представленные цветовой картой; цвета даны в топографической шкале, где синий цвет соответствует малым значениям, когда коричневый/красный цвета соответствуют большим значениям интенсивности

7.5.1. Данные

В трехмерном наборе данных эмбриона зебрафиш каждая точка данных соответствует ядру, каждое из которых представлено массивом чисел: три пространственные координаты центроида ядра и интенсивность флуоресценции (в произвольных единицах) меченых генов (обычно на один эмбрион приходится два гена). Геометрически точки данных распределены вокруг трехмерного эллипсоида в несколько нерегулярных слоев (см. рис. 7.26, 7.29а, 7.30а). Если представить яйцо рыбы в виде эллипсоида (или сфера), то ранний эмбрион рыбы может быть геометрически описан как толстая (многослойная) сферическая “шапочка”, накрывающая яйцо, которая может быть спрятана до диска без существенных искажений по краям (биологи называют геометрию этих эмбриональных стадий “куполом” или “диском”). Ключевые гены, изучаемые на этих ранних стадиях, обычно формируют паттерны экспрессии в компактных подобластях сферического колпачка, таких как открытые кольца и т.д. Процедуры предварительной обработки и SSA могут быть сфокусированы или ограничены этими подобластями. Другими словами, мы предполагаем, что существует преобразование данной области экспрессии в параллелепипед и что это преобразование не искажает данные кардинально. Если область экспрессии слишком велика для такого преобразования в целом, то преобразование может быть выполнено для нескольких независимых частей области.

7.5.2. Метод

Методы типа SSA обрабатывают данные, заданные на регулярной сетке внутри параллелепипеда. Поэтому при применении к нерегулярным 3D данным сначала требуется спрям-

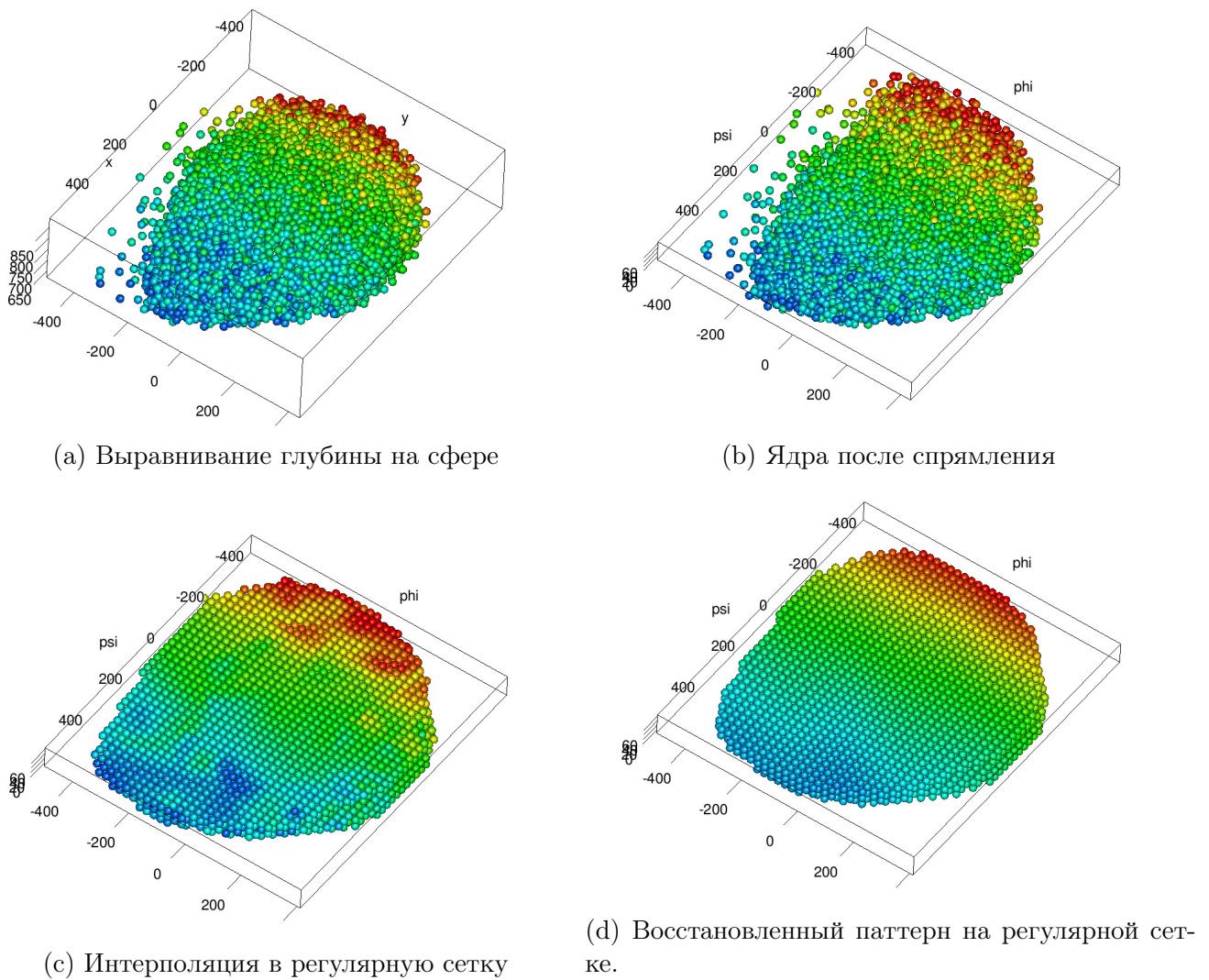


Рис. 7.27: Шаги обработки в подходе 3D-SSA.

ление, а затем регуляризация.

Обработка 3D данных, которые находятся в слое около эллипсоидальной поверхности, состоит из следующих шагов:

- Определение местоположения данных: оценка центра эллипсоида и нахождение положения центроидов ядер относительно него. Возможно, добавление вращения данных для более простого, не искажающего спрямления.
- Спрямление данных и встраивание их в параллелепипед.
- Интерполяция на регулярную сетку $\{i = 1, \dots, N_1\} \times \{j = 1, \dots, N_2\} \times \{k = 1, \dots, N_3\}$ для получения f_{ijk} .
- Применение 3D-SSA, возможно в версии Shaped, чтобы можно было анализировать 3D подобласти. 3D-SSA приводит к разложению вида: $f_{ijk} = s_{ijk} + r_{ijk}$, где s_{ijk} соответствует паттерну экспрессии, или тренду.

- Интерполяция s_{ijk} обратно к ядрам в параллелепипеде.
- Преобразование обратно к исходным координатам ядер в эмбрионе.

В результате этого процесса получается извлеченный паттерн и остаточный шум в исходной геометрии эмбриона. Это позволяет продолжить изучение формы паттерна (например, сравнение с детерминированными динамическими моделями регуляции генов), а также модели шума (например, определить, является ли шум аддитивным или мультипликативным).

Ниже мы более подробно прокомментируем этапы схемы обработки. Для простоты мы предполагаем, что данные расположены вблизи поверхности сферы (как в случае с яйцами зебрафиш).

Определение местоположения данных

Начало системы координат — центр сферы, оцениваемый как точка, наиболее равнодistantная от всех точек данных (более формально, мы находим точку, минимизирующую дисперсию расстояний между ее положением и положением точек данных).

Рассматриваются два типа распределений пространственных данных, для каждого из которых существует своя процедура переориентации и спрямления.

Первый тип относится к данным, расположенным вблизи сферической крышки. В этом случае ось ‘z’ проходит через центр сферы и среднюю точку данных; мы можем повернуть данные, чтобы получить положительные значения ‘z’ для всех координат ядер.

Во втором типе данные расположены в полосе вдоль экватора сферы. В этом случае ось ‘z’ выбирается ортогональной к экваториальной плоскости и проходящей через центр сферы.

В обоих случаях оси ‘x’ и ‘y’ выбираются ортогональными к оси ‘z’ и друг к другу, ориентированными так, чтобы максимально сохранить исходную ориентацию осей.

Спрямление данных

Данные вписываются в параллелепипед с гранями, параллельными осям. Первая ось соответствует глубине слоя данных, а вторая и третья оси — направлениям на поверхности. При спрямлении мы стремимся сохранить пропорции данных как можно более неизменными.

Выравнивание данных по глубине Мы предполагаем, что данные расположены в слое с примерно постоянной глубиной на сферической поверхности. Перед проектированием данные должны быть скорректированы, чтобы они находились в идеально-сферическом слое постоянной глубины. Предположим, что все ядра расположены в области, которая ограничена двумя выпуклыми поверхностями (например, в сферическом слое).

Первым шагом процедуры является нахождение этих поверхностей как выпуклых внешних и внутренних оболочек, применяя классический метод “выпуклой оболочки” к исходным данным для нахождения внешней оболочки и к инвертированным данным для нахождения

внутренней. Затем найденные внешняя и внутренняя оболочки преобразуются в сферические поверхности радиусов $R + D/2$ и $R - D/2$ соответственно, где радиус сферы R оценивается как медиана расстояний от точек данных до центра сферы, которая была оценена в разделе 7.5.2, а глубина слоя D оценивается как медианное расстояние между оболочками.

Процедура может обнаружить слишком тонкий слой глубиной в одно ядро, когда следует использовать 2D-SSA вместо 3D версии. Для данных, которые являются двумерными, применяется метод 2D-SSA, рассмотренный в разделе 5.2.3 В противном случае разрабатывается новый модифицированный 3D подход.

Проекция Сферические проекции могут иметь различные инварианты. Для данных в сферических шапках мы используем эквидистантную азимутальную проекцию [209, Раздел 25], чтобы сохранить расстояние от центральной точки (т.е. широту) и, следовательно, линейные размеры исходного слоя. После спрямления и проектирования, точки данных (ядра) будут иметь следующие координаты: ψ, φ — координаты вдоль ортогональных меридианов в азимутальной проекции, d — глубина; все измеряются в метрических единицах.

Для данных в экваториальных полосах мы используем эквидистантную цилиндрическую проекцию [209, Раздел 12], сохраняя расстояния точек до экватора (т.е. широты) и расстояния между точками на экваторе. Это дает аналогичные координаты ψ, φ и d (ширина, долгота и глубина, измеренные в метрических единицах). Если экваториальная полоса огибает весь экватор, то мы получаем параллелепипед с круговой топологией по экваториальной координате и можем применить круговую версию SSA [41].

Обратим внимание, что мы получаем новые координаты примерно в тех же единицах (и в тех же пропорциях), что и исходные данные. Это основная цель использования эквидистантных проекций, так как пропорции могут сильно повлиять на интерполяцию к регулярной сетке.

Мы используем относительные координаты для спрямленных данных. Так, на всех рисунках d представлено в процентах, с 0% на внутренней поверхности и 100% на внешней; ψ, φ представлены как доли длины экватора.

Интерполяция ядер в регулярную сетку и обратно

Интерполяция нерегулярных данных в регулярную сетку известна как ‘интерполяция 3D рассеянных данных’ (см. [210] для описания и обзора различных подходов).

Мы используем подход “линейная интерполяция на основе триангулированной нерегулярной сети с использованием триангуляции Делоне”, где интерполяция строится путем линейной интерполяции значений вершин из соответствующего симплекса триангуляции. Реализация осуществляется с помощью библиотеки ‘CGAL’ [211].

Обратим внимание, что ядра не обязательно заполняют весь параллелепипед, а также учитываются краевые эффекты. Поэтому после интерполяции мы получаем данные об экс-

прессии на подмножестве сетки параллелепипеда. Например, для данных о форме шапочки подмножество представляет собой диск. Это подмножество можно обработать с помощью Shaped версии 3D-SSA.

Для обратной интерполяции к ядрам используется обычная трилинейная интерполяция для восстановления паттерна. Остаточные значения рассчитываются как разность между исходными значениями экспрессии и значениями экспрессии, полученными при извлечении паттерна (тренда).

Shaped 3D-SSA

Методы Shaped 3D-SSA требуют, чтобы исходные данные были представлены на подмножестве регулярной сетки (которую мы называем начальной формой).

Параметром метода является фигурное 3D окно, вписанное в параллелепипед размером $L_1 \times L_2 \times L_3$.

Предполагается, что выбранное окно покрывает все точки исходной формы. Если это не так, то непокрытые точки сетки удаляются. Для правильной формы окна количество удаленных точек не должно быть слишком большим, иначе следует выбрать другую форму или размер окна.

В разделе 5.1 метод Shaped SSA описан для 2D-объектов, для простоты обозначений. В этом разделе мы рассматриваем метод Shaped 3D-SSA, который образуется путем естественного переноса Shaped 2D-SSA на трехмерный случай.

Выбор параметров

Разложение данных экспрессии генов при нерегулярном расположении позиции ядер имеет несколько параметров: для интерполяции и спрямления; и для SSA (форма и размер окна, а также число r компонентов для восстановления).

Для того чтобы получить достаточно большое количество точек сетки по отношению к количеству ядер, шаги регулярной сетки не должны быть слишком большими. Верхняя граница для точек сетки ограничена только вычислительными затратами.

Рекомендации для 3D-SSA аналогичны рекомендациям для 2D- и 1D-SSA, приведенным в предыдущих главах. Большие окна соответствуют более тонкому разложению и более точному восстановлению, если сигнал (паттерн) имеет простую структуру, генерирующую несколько SVD-компонентов. Для более сложных паттернов предпочтительны средние и маленькие окна.

Обратим внимание, что размер окна измеряется относительно особенностей паттерна и не должен зависеть от шага интерполяции. Поэтому размеры окон измеряются в процентах от размеров изображения, а не в количестве точек сетки. Начальный размер окна может быть выбран равным примерно 10-20% от размера изображения в каждом направлении. Если

паттерн извлечен неточно, окно может быть увеличено; если паттерн смешан с остатком (шумом), окно должно быть уменьшено.

Идентификация компонентов паттерна может быть выполнена путем анализа форм собственных массивов или элементарных восстановленных компонентов (см. пример в разделе 5.3). Медленно меняющиеся паттерны могут быть легко построены путем накопления медленно меняющихся элементарных компонентов. Поскольку визуальный анализ трехмерных объектов затруднен, предпочтительнее изображать срезы (одномерные графики или двухмерные изображения), полученные путем фиксации одной или двух координат.

Качество извлечения паттерна можно проверить по поведению остатков. Для правильного выделения паттерна остатки должны колебаться около нуля. Поэтому рекомендуется выбирать медленно меняющиеся элементарные компоненты, чтобы остатки не попадали в паттерн.

Модель остатков

Для понимания биологических источников шума в экспрессии генов, представляет интерес извлечение из данных модели шума. Мы предлагаем метод обнаружения модели шума, основанный на стандартном teste на гетероскедастичность остатков с различными нормализациями.

Для разложения исходных данных на паттерн и шум: $x_i = s_i + r_i$, $i = 1, \dots, N$, где N — число ядер (перечисление по одному индексу вместо трех не влияет на результаты), предположим, что шум в ядрах независим и рассмотрим модель

$$r_i = \varepsilon_i \cdot |s_i|^\alpha, \quad (7.11)$$

где $E\varepsilon_i = 0, D\varepsilon_i = \text{const}$.

Если $\alpha = 0$, то шум аддитивный (его стандартное отклонение не зависит от значений паттерна). Если $\alpha = 1$, то шум мультипликативный (его стандартное отклонение пропорционально значениям паттерна). Промежуточное значение $\alpha = 0.5$ соответствует пуассоновскому шуму, где дисперсия шума пропорциональна величине паттерна.

Метод Парка [212] оценивает α как наклон линейной регрессии в модели

$$\log |r_i| = \log \sigma + \alpha \log |s_i| + v_i, \quad (7.12)$$

где $v_i = \log |\varepsilon_i/\sigma|$ является подходящим членом ошибки. Метод Парка оказывается устойчивым к распределению остатков. Оценка α в модели (7.12) может быть получена, например, методом наименьших квадратов.

Реализация

Все описанные методы реализованы с помощью в R и включены в пакет BioSSA.

7.5.3. Пример для модели расположении ядер на части сферы (“шапочке”)

Здесь мы рассмотрим применение процедуры 3D-SSA на близком к сферическому яйце зебрафиш. Координаты ядер взяты из стандартного примера данных в MatchIT [206]. Инструмент ‘MatchIT’ был запущен с набором данных и параметрами по умолчанию, в результате чего были получены файлы с автоматически обнаруженными ядрами. Обработке подвергся файл ‘gsc_ntla_wt_t008_ch01.csv’, содержащий координаты ядер для клеток, экспрессирующих ген *gsc* (‘goosecoid’) на стадии развития ‘late shield’.

В этих данных ядра расположены толстым слоем на сфере. Каждое ядро обозначено ‘1’ или ‘0’ для гена, экспрессирующего или нет, соответственно. Для этого набора данных *gsc*-экспрессирующие ядра (‘1’) расположены в пределах небольшой сферической шапочки. Мы расширяем анализируемую область, чтобы включить все ‘1’-ядра, плюс окружающее кольцо ядер.

Сначала мы представим результаты для обоих примеров и объясним выбор параметров и компонентов паттерна. Затем мы показываем, как метод может быть использован для оценки модели шума после извлечения паттерна. В рассматриваемых примерах мы добавляем гауссовский шум к паттернам, однако это не обязательно, так как SSA-семейство методов устойчиво к распределению шума и возможным слабым зависимостям.

Двухэкспоненциальный паттерн экспрессии

Давайте построим значения экспрессии следующим образом:

$$s(\psi, \varphi, d) = e^{2\varphi + 9\psi + 2d} + 2e^{-(2\varphi + 13\psi + d)}, \quad (7.13)$$

$$v(\psi, \varphi, d) = s(\psi, \varphi, d) + \varepsilon \cdot s(\psi, \varphi, d), \quad (7.14)$$

где ψ , φ — относительные сферические координаты, d — глубина слоя ($d \in [0, 1]$) и ε — белый гауссовский шум со стандартным отклонением 0.35. Этот паттерн, который мы называем CAP-2EXP, представляет собой сумму двух экспонент плюс мультипликативный шум. Обратим внимание, что паттерн зависит от трех координат и поэтому изменяется в трех направлениях. Умеренные уровни шума здесь и в других рассмотренных имитационных примерах были выбраны для лучшего визуального цветового представления результатов для рассматриваемых паттернов.

Внутренний и внешний виды оболочек области ядер показаны на рис. 7.28a–7.28b, где цвета соответствуют уровням экспрессии, как на рис. 7.26. Видно, что окраска пестрая (ядра разного цвета занимают близкие позиции), что отражает наличие шума.

Подход, описанный в разделе 7.5.2, дает восстановленный паттерн, изображенный на рис. 7.28c и 7.28d. Хорошо видны результаты удаления шума.

Для лучшего визуального представления можно изобразить сами ядра, см. рис. 7.29 и 7.30. Как и раньше, цвет ядра отражает уровень экспрессии в той же шкале, что и на

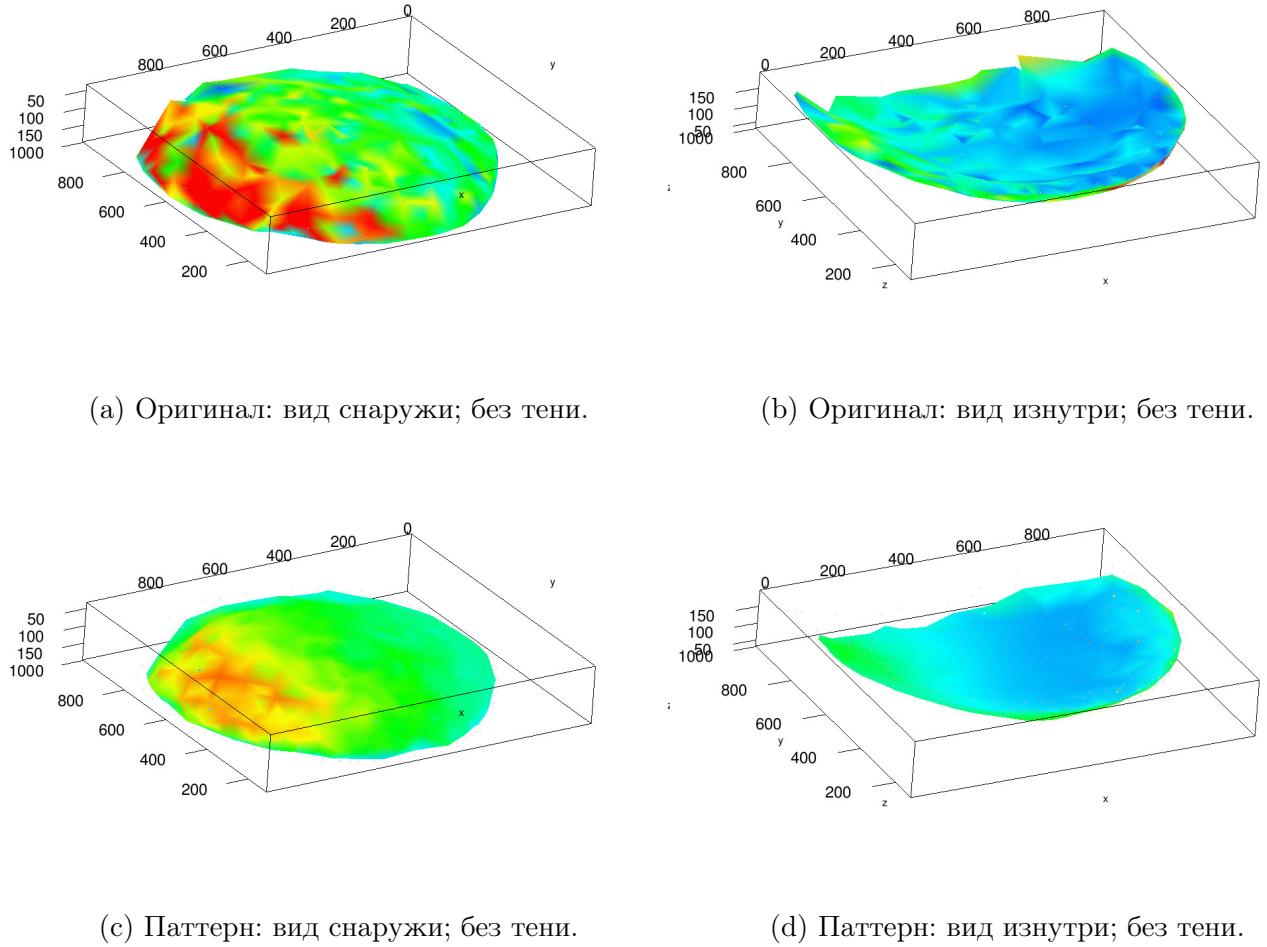


Рис. 7.28: CAP-2EXP: Оболочки области ядер, окраска на основе интенсивности экспрессии оригинала и паттерна.

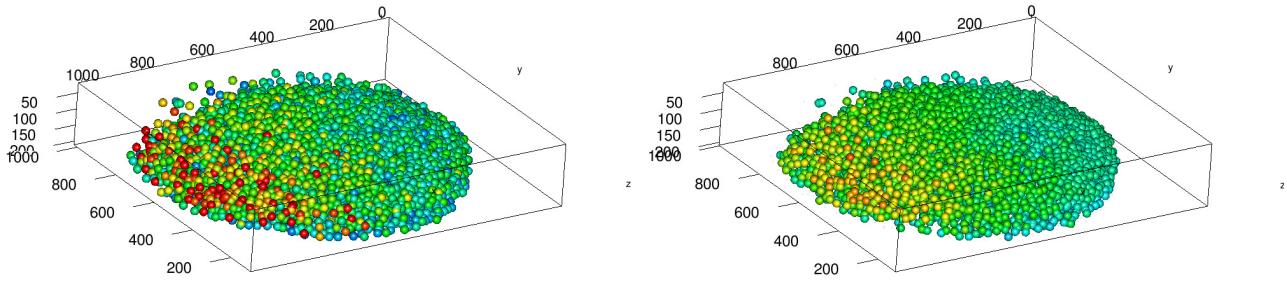
рис. 7.26. Характер экспрессии можно четко проследить в очищенных от шума данных. Разница между внутренним и внешним видом демонстрирует динамику паттерна в направлении глубины.

После извлечения паттерна можно оценить модель шума (см. раздел 7.5.2). В модель (7.13) был включен мультипликативный шум с $\alpha = 1$. Применение метода Парка дает оценку $\hat{\alpha} = 1.073$, восстанавливая мультипликативный характер сгенерированного шума и демонстрируя, как метод обработки данных может различать, например, аддитивный, пуассоновский и мультипликативный шум в наборах данных.

Выбранные параметры

После спрямления, которое описано в разделе 7.5.2, получаем параллелепипед с длиной $l \approx 1030$, шириной $w \approx 885$ и глубиной $d \approx 50$. Поскольку форма спрямленного облака ядер похожа на сферическую шапку, была применена равноудаленная азимутальная проекция.

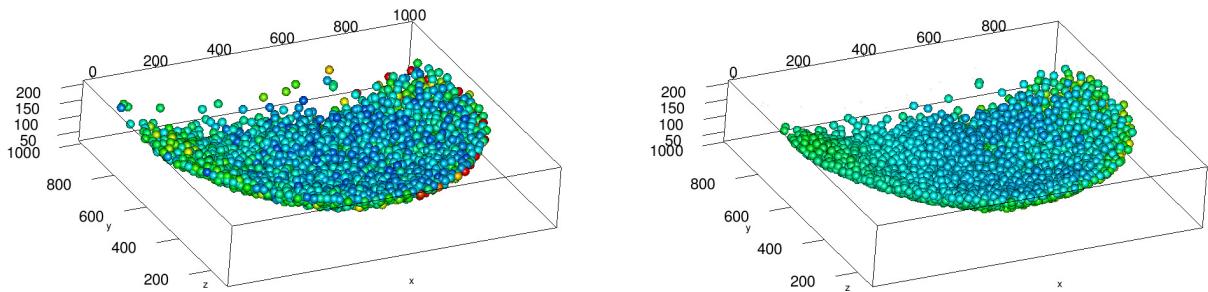
Размер шага был выбран одинаковым во всех направлениях для получения 10^6 точек



(a) Оригинальная интенсивность экспрессии.

(b) Восстановленный паттерн.

Рис. 7.29: CAP-2EXP: ядра окрашены; вид снаружи.



(a) Оригинальные интенсивности экспрессии.

(b) Восстановленный паттерн.

Рис. 7.30: CAP-2EXP: ядра окрашены; вид изнутри.

сетки.

При применении Shaped 3D-SSA эллипсоидное окно было вписано в параллелепипед размером $L_1 \times L_2 \times L_3$, где L_i равны 40% от размеров исходного изображения. Общее количество ядер в файле данных составляет 3595, а выбранное окно охватывает в среднем около 160 ядер. Количество ядер, охваченных всеми позициями выбранного окна, составляет 3306, то есть несколько боковых ядер не учитывались.

Для выявления компонентов паттерна рассмотрим 2D срезы 3D собственных массивов и элементарные восстановленные компоненты. Первые два эллипсоидальных собственных массива гладкие, третий имеет некоторые осцилляции. Поэтому мы выбираем первые две компоненты для восстановления паттерна.

Шесть ведущих элементарных компонент, которые генерируются шестью собственными массивами, вместе с исходными и остаточными 2D срезами показаны на рис. 7.31.

Рис. 7.31 подтверждает выбор двух ведущих компонент (то есть $r = 2$) как соответствующих паттерну.

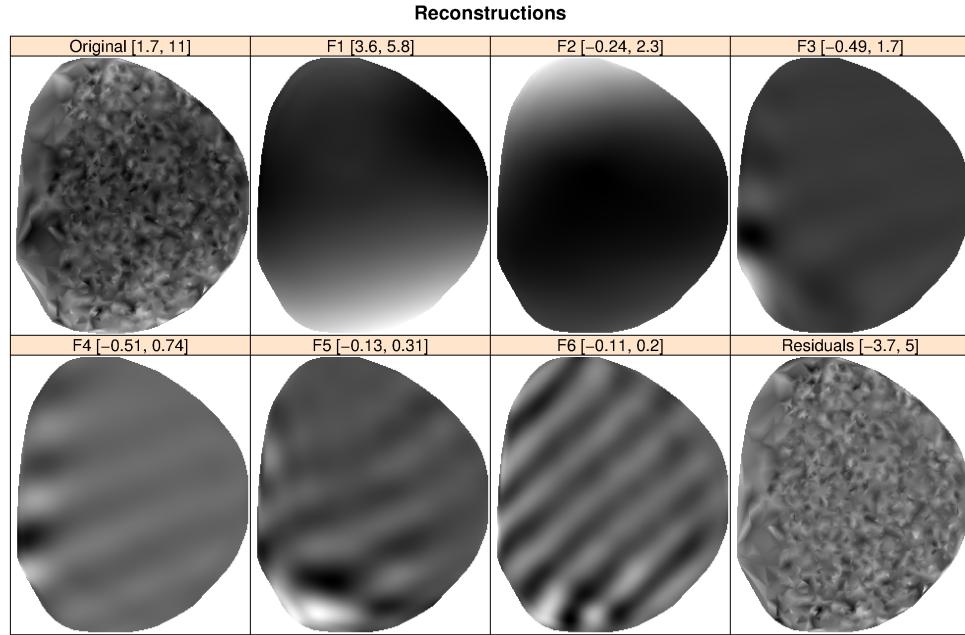


Рис. 7.31: CAP-2EXP: Восстановление элементарных компонентов, срезы.

Паттерн восстанавливается из главных собственных массивов на регулярных точках сетки, затем интерполируется обратно к спрямленным (нерегулярным) ядрам, а затем преобразуется в исходные позиции ядер на яйце зебрафин.

Результат восстановления достаточно устойчив к выбору окна. Например, восстановление паттерна будет примерно таким же, если мы выберем размер окна 35% вместо 40% от исходного размера. Однако для более сложных паттернов предпочтительнее меньшие размеры окна; см. обсуждение выбора окна в [29] (1D случай) и [208] (2D случай).

Проверка правильности выделения паттерна

Осмотр восстановленных изображений ясно демонстрирует удаление шума для обоих тестовых паттернов. Однако это не доказывает, что мы восстановили весь паттерн. Теперь мы покажем на примере CAP-2EXP, что восстановление паттерна завершено.

Рассмотрим двумерные срезы восстановленных значений на регулярной сетке после спрямления расположения, фиксируя глубину. Вертикальная ось будет представлять значения экспрессии (цветовое отображение, например, на рис. 7.28), а горизонтальные оси соответствуют ϕ , ψ координатам ядер или узлов сетки.

Поскольку ядра могут быть расположены не точно на срезе, мы рассматриваем ядра из слоя плюс минус 10% в каждую сторону. Выделенный паттерн на срезе по ядрам изображается сплошной поверхностью, а значения экспрессии ядер показаны отдельными точками,

см. рис. 7.32 (вверху). На рис. 7.32 (внизу) остатки получены путем вычитания паттерна из значений на ядрах. Равномерный разброс остатков вокруг нулевой ‘xy’-плоскости указывает на хорошее соответствие восстановления данным.

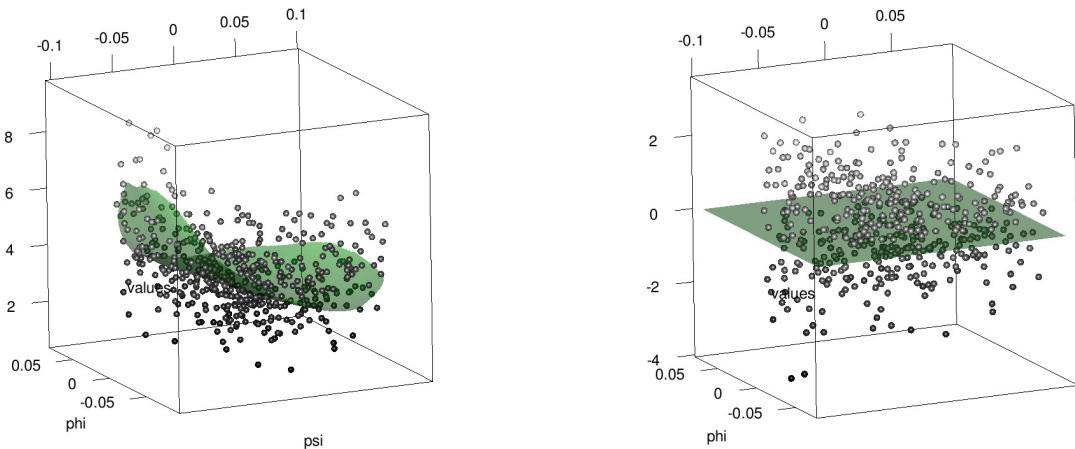


Рис. 7.32: CAP-2EXP паттерн: 2D срез со значениями экспрессии, глубина $d = 82.5\%$; вверху: поверхность изображает восстановленный паттерн (в точках регулярной сетки), точки показывают значения экспрессии в отдельных ядрах из слоя $\pm 10\%$; внизу: значения остатки от восстановления на ядрах равномерно рассеяны вокруг нулевой плоскости

Для более тонкого анализа мы можем построить одномерные срезы. Например, зафиксировав глубину на 82,5% и ширину (φ) на 50%, мы рассматриваем двухэкспоненциальный паттерн для ядер из тонкого слоя ($\pm 10\%$) вокруг этой глубины и ширины. Выделенный паттерн на срезе по ядрам изображен сплошной линией. Рис. 7.33 подтверждает качество восстановления, остатки между данными и восстановлением равномерно распределены вокруг нулевой плоскости. Заметим, что для оценки модели шума выбор 1, 2, 3 или 4 ведущих компонент мало влияет на результаты.

Таким образом, мы продемонстрировали, что 3D-SSA способен извлекать трехмерные паттерны из данных экспрессии генов, представленных на нерегулярной сетке, а затем анализировать структуру биологического шума.

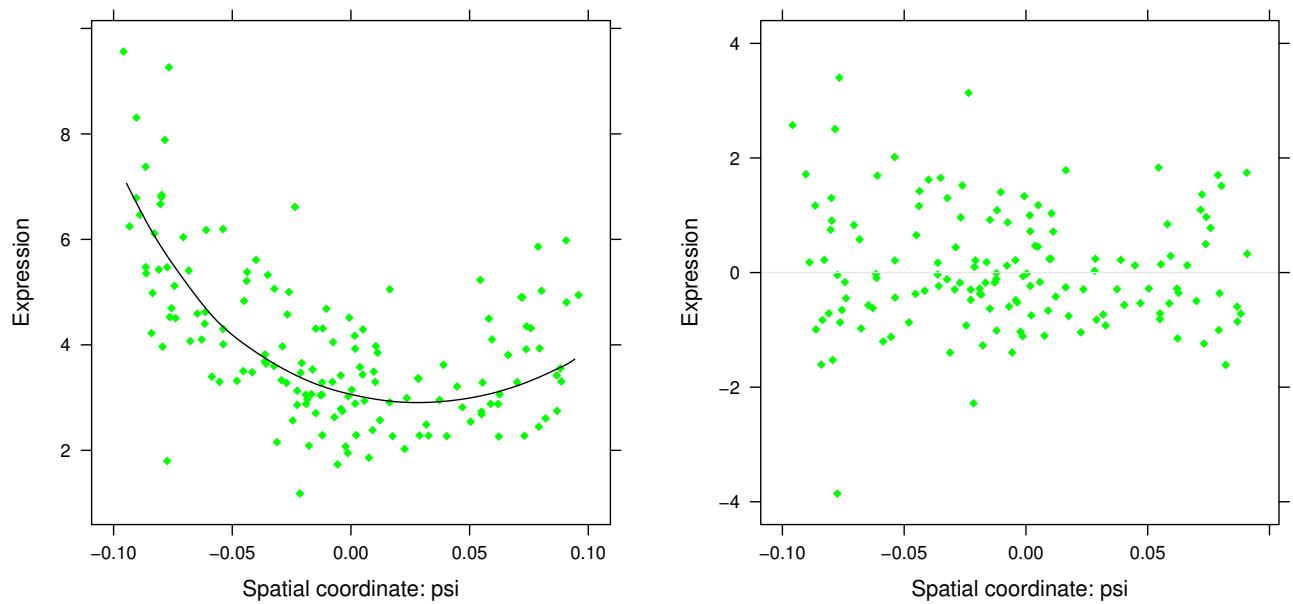


Рис. 7.33: Паттерн CAP-2EXP: одномерный срез на глубине 82,5% (d) и ширине 50% (φ). Слева: схема на сетке (кривая) и исходные значения на ядрах (точки). Справа: остатки между восстановлением и значениями на ядрах.

Заключение

В работе были рассмотрены различные аспекты метода анализа сингулярного спектра. Опишем основные полученные результаты и коротко их обсудим.

Одним из базовых результатов является построение общей схемы семейства методов SSA для разложения исходного объекта (например, временного ряда или цифрового изображения) на интерпретируемые составляющие (глава 1). Получилась схема, в которой можно зафиксировать тип объекта и менять способ построения разложения на элементарные компоненты для более точного результата, а можно, наоборот, зафиксировать тип разложения и применять его к объектам разной размерности и формы. Такой подход позволяет развивать теорию в едином стиле, а также разрабатывать реализации предлагаемых алгоритмов в единой структуре.

Вторая важная тема относится к этапу разложения общей схемы SSA. Успешность разложения временного ряда на элементарные компоненты зависит от того, можно ли их потом сгруппировать так, чтобы выделить из ряда нужные компоненты, например, тренд или периодику. В существующей базовой версии SSA строящееся разложение обладало рядом недостатков (хотя и было оптимальным среди всех адаптивных разложений). Например, две синусоидальные компоненты было не разделить, если у них были одинаковые амплитуды. Было доказано, что методы DerivSSA и SSA-ICA справляются с этой проблемой. Другой пример: для отделения тренда от периодики в коротком временном ряде в базовом варианте накладываются довольно жесткие условия на длину ряда и длину окна. Метод Iterative OSSA позволяет разделить такие компоненты. В целом, после появления предложенных методов круг отделимых компонент значительно расширился. Улучшению отделимости полиномиального тренда помогает метод ProjSSA. Все эти методы описаны в главе 2 для одномерных временных рядов. Заметим, что они естественным образом расширяются на многомерные объекты. Мы не выписывали результаты для многомерных исходных данных, так как алгоритмы аналогичные, а запись была бы слишком громоздкой. В R-пакете RSSA включены реализации многомерных методов DerivSSA и Iterative OSSA.

Результаты главы 3 охватывают другой аспект применения SSA к одномерным временным рядам. А именно, рассматривается случай сигнала, управляемого линейной рекуррентной формулой. Были предложены и обоснованы алгоритмы для заполнения пропусков во временных рядах. Заметим, что предлагаемый подход через заполнения пропусков части компонент векторе из заданного подпространства, являющейся основной при заполнении пропусков во временном ряде, естественным образом расширяется и на заполнение пропусков в многомерных объектах. Как и для прогноза, в случае заполнения пропусков хотя теоретическое обоснование и подразумевает, что сигнал управляетяя ЛРС, но построенные методы применимы и для случая, когда это верно только приближенно. Так как класс рядов, управляемых ЛРС, состоит из сумм произведений многочленов, экспонент и гармоник, то формально на конечном промежутке практически любую последовательность можно приблизить рядом

из данного класса; вопрос только в ранге аппроксимирующего ряда, который должен быть небольшим.

Другой результат главы 3 касается случая, когда сигнал должен в точности управляться ЛРС небольшой размерности, что встречается, как правило, в инженерных задачах. В этом случае задачу можно описать как взвешенную задачу наименьших квадратов, где веса подстроены под автоковариационную матрицу авторегрессионного шума. При этом, в различных статьях предлагаются как матричные, так и векторные постановки задачи. Было доказано, что для случая авторегрессионного шума в матричной форме невозможно задать оптимальные веса (т.е. веса, приводящие к асимптотически по длине ряда минимальной дисперсии оценки сигнала). Доказательство для случая авторегрессии порядка 1 и 2 является полным, а для общего случая доказан только базовый результат. Полученный результат не означает, что довольно простое решение задачи в матричной форме неприменимо. Следствием полученного результата является то, что в матричной постановке задачи веса должны подбираться численно для того, чтобы они были как можно ближе к оптимальным.

В главу 3 были включены результаты по обнаружению сигнала в красном шуме. В отличие от других задач, эта задача довольно специфична, как требованием наличия именно красного шума, так и лишь частичным участием алгоритма SSA. Однако, уже известный алгоритм называется Монте-Карло SSA, поэтому мы его рассмотрели наряду с другими методами. В рамках подхода Монте-Карло SSA был предложен множественная взвешенная версия алгоритма для проверки гипотезы о том, что сигнала в шуме нет, которая была статистически обоснована с точки зрения контроля групповой ошибки первого рода. Полученный результат может быть обобщен на случай MSSA, однако многомерный случай гораздо сложнее с точки зрения наличия разных сигналов в разных рядах из анализируемой системы временных рядов. Поэтому обобщение требует отдельного исследования.

Также глава 3 содержит исследование влияния длины окна на точность оценки сигнала и прогноза. Исследование в основном численное, хотя и содержит результат для случая константного сигнала, полученный аналитически.

Часть диссертационной работы посвящена многомерным обобщениям SSA. Наиболее часто встречающимся обобщением является метод MSSA для анализа системы временных рядов, так как есть представление, что при некоторых условиях точность выделения сигнала может стать выше, чем анализ каждого временного ряда отдельно. Введено понятие согласованных временных рядов, которое является ключевым для улучшения точности, а также выписаны достаточные условия разделимости для компонент ряда. Материал, касающийся MSSA, был выделен в отдельную главу 4, так как, с одной стороны, MSSA является самым близким расширением SSA, а, с другой стороны, имеет свои существенные особенности.

Следующий момент, который хочется отметить, — это разработка общего подхода к объектам разной размерности и формы через подход, называемый Shaped SSA. В главе 5 предложена формализация, которая позволяет в едином стиле рассматривать все варианты объектов в одномерном и многомерном случае. В качестве частных случаев приведен анализ

многомерных цифровых изображений, а также показано, что как SSA анализ рядов с пропусками, так и MSSA можно рассматривать как частные случаи Shaped SSA. Подход через Shaped SSA полезен как с точки зрения унификации, так и с точки зрения анализа цифровых изображений непрямоугольной формы. В качестве примера был рассмотрено фото Марса, который, имеет круглую форму, соответственно, окно было тоже кругом.

В главах 2, 4, and 5 содержится также описание подхода к автоматической идентификации элементарных компонент в разложении для выделения тренда и периодик. Подход основан на аналогичных принципах, однако имеет свою специфику для разных видов данных.

Описанные выше общие подходы к алгоритмам SSA и их модификациям позволили создать пакет RSSA, написанный на языке R. Пакет содержит в себе очень много разных методов, применимых к объектам разной формы и, пожалуй, его создание было бы невозможно без развития общих подходов, описанных выше. Структура пакета описана в главе 6.

Работу завершает глава 7, посвященная приложениям метода SSA к реальным данным. Первое приложение состоит в применении SSA к прогнозу параметров вращения Земли, где в сравнении с публично выкладываемыми прогнозами из двух других источников было получено преимущество SSA. В другом примере способность SSA сглаживать временные ряды была использована для анализа одномерных данных, не имеющих отношения к временным рядам, а именно, для оценки плотности по эмпирическим данным. Оказалось, что подход через сглаживание эмпирической функции распределения, особенно в случае искусственно дискретных данных, дает хороший результат.

Участие в научных проектах, посвященных анализу экспрессии генов, привело к тому, что три раздела посвящены именно этой теме. Заметим, что данные также не являются временными рядами; они получены измерением активности генов в разных пространственных точках. В разделе 7.3 применение SSA для оценки параметров предложенной модели позволило рассмотреть динамику развития эмбриона количественно, через изменение параметров модели. В двух следующих разделах показаны примеры применения 2D-SSA и 3D-SSA для выделения сигнала в случае двумерных и трехмерных данных экспрессии генов. Одно из применений результатов — построение модели шума, так как до сих пор нет определенного мнения, шум аддитивный или величина шума пропорциональна величине паттерна. Метод 2D-SSA хорошо справляется с отделением шума и имеет средства проверки, насколько хорошо выделен паттерн, однако биологические выводы отсложняются недостаточным качеством данных, наличием так называемого фона при измерениях. Другой эффект, который удалось найти и устраниить с помощью 2D-SSA — это эффект смешивания активности разных генов при одновременном измерении. Успешность применения опирается, в том числе, на то, что можно выделить интересующие области не обязательно прямоугольной формы и применить к ним Shaped 2D-SSA.

Применение SSA к трехмерным данным является в большой степени иллюстративным, так как на данный момент доступно очень мало таких реальных данных и поэтому анализи-

ровались частично искусственные данные. Однако, разработанная методика анализа активности генов в не равноотстоящих трехмерных точках представляется полезной как пример применения к данным, который при развитии измерительной техники в будущем должны появляться в большем количестве.

Благодарности

Я благодарна моим коллегам с кафедры статистического моделирования Санкт-Петербургского государственного университета за помощь, моему научному руководителю и затем коллеге Владимиру Викторовичу Некруткину за то, что научил меня отношению к работе и жизни, моим бакалаврам, магистрантам и аспирантам за мотивацию и разработку методологии SSA вместе со мной. Последнее, но не менее важное: вся благодарность и любовь моей семьи, мужу и дочери за поддержку, а внучке — за хорошее настроение.

Список литературы

1. *Golyandina N., Nekrutkin V., Zhigljavsky A.* Analysis of Time Series Structure: SSA and Related Techniques. — Chapman&Hall/CRC, 2001.
2. *Markovsky I.* Low Rank Approximation: Algorithms, Implementation, Applications (Communications and Control Engineering). — 2nd edition. — Springer, 2019.
3. *Cadzow J. A.* Signal enhancement: a composite property mapping algorithm // *IEEE Transactions on Acoustics, Speech, and Signal Processing*. — 1988. — Vol. 36, no. 1. — Pp. 49–62.
4. *Golyandina N., Osipov E.* The “Caterpillar”-SSA method for analysis of time series with missing values // *Journal of Statistical Planning and Inference*. — 2007. — Vol. 137, no. 8. — Pp. 2642–2653.
5. *Golyandina N.* On the choice of parameters in Singular Spectrum Analysis and related subspace-based methods // *Statistics and Its Interface*. — 2010. — Vol. 3, no. 3. — Pp. 259–279.
6. Gene expression noise in spatial patterning: Hunchback promoter structure affects noise amplitude and distribution in Drosophila segmentation / D.M. Holloway, F.J.P. Lopes, L. da Fontoura Costa et al. // *PLoS Computational Biology*. — 2011. — Vol. 7, no. 2.
7. *Golyandina N., Pepelyshev A., Steland A.* New approaches to nonparametric density estimation and selection of smoothing parameters // *Computational Statistics and Data Analysis*. — 2012. — Vol. 56, no. 7. — Pp. 2206–2218.
8. *Golyandina N., Korobeynikov A.* Basic Singular Spectrum Analysis and forecasting with R // *Computational Statistics and Data Analysis*. — 2014. — Vol. 71. — Pp. 934–954.
9. *Golyandina N., Shlemov A.* Variations of singular spectrum analysis for separability improvement: Non-orthogonal decompositions of time series // *Statistics and Its Interface*. — 2015. — Vol. 8, no. 3. — Pp. 277–294.
10. Shaped singular spectrum analysis for quantifying gene expression, with application to the early Drosophila embryo / A. Shlemov, N. Golyandina, D. Holloway, A. Spirov // *BioMed Research International*. — 2015. — Vol. 2015.
11. Shaped 3D Singular Spectrum Analysis for Quantifying Gene Expression, with Application to the Early Zebrafish Embryo / A. Shlemov, N. Golyandina, D. Holloway, A. Spirov // *BioMed Research International*. — 2015. — Vol. 2015.
12. Multivariate and 2D extensions of singular spectrum analysis with the Rssa package / N. Golyandina, A. Korobeynikov, A. Shlemov, K. Usevich // *Journal of Statistical Software*. — 2015. — Vol. 67, no. 2.
13. *Zhigljavsky A., Golyandina N., Gryaznov S.* Deconvolution of a discrete uniform distribution // *Statistics and Probability Letters*. — 2016. — Vol. 118. — Pp. 37–44.
14. *Golyandina N.E., Lomtev M.A.* Improvement of separability of time series in singular spectrum analysis using the method of independent component analysis // *Vestnik St.*

Petersburg University: Mathematics. — 2016. — Vol. 49, no. 1. — Pp. 9–17.

15. Zvonarev N., Golyandina N. Iterative algorithms for weighted and unweighted finite-rank time-series approximations // *Statistics and Its Interface*. — 2017. — Vol. 10, no. 1. — Pp. 5–18.
16. Golyandina N., Shlemov A. Semi-nonparametric singular spectrum analysis with projection // *Statistics and Its Interface*. — 2017. — Vol. 10, no. 1. — Pp. 47–57.
17. Two-exponential models of gene expression patterns for noisy experimental data / T. Alexandrov, N. Golyandina, D. Holloway et al. // *Journal of Computational Biology*. — 2018. — Vol. 25, no. 11. — Pp. 1220–1230.
18. Zhornikova P., Golyandina N., Spirov A. Noise model estimation with application to gene expression // *Journal of Bioinformatics and Computational Biology*. — 2019. — Vol. 17, no. 2.
19. Golyandina N., Zhigljavsky A. Blind deconvolution of covariance matrix inverses for autoregressive processes // *Linear Algebra and Its Applications*. — 2020. — Vol. 593. — Pp. 188–211.
20. Golyandina N. Particularities and commonalities of singular spectrum analysis as a method of time series analysis and signal processing // *Wiley Interdisciplinary Reviews: Computational Statistics*. — 2020. — Vol. 12, no. 4.
21. Quantification reveals early dynamics in Drosophila maternal gradients / A. Shlemov, T. Alexandrov, N. Golyandina et al. // *PLoS ONE*. — 2021. — Vol. 16, no. 8 August.
22. Zvonarev N., Golyandina N. Fast and stable modification of the Gauss–Newton method for low-rank signal estimation // *Numerical Linear Algebra with Applications*. — 2022. — Vol. 29, no. 4.
23. Zvonarev N., Golyandina N. Mixed Alternating Projections with Application to Hankel Low-Rank Approximation // *Algorithms*. — 2022. — Vol. 15, no. 12. — P. ID 460.
24. Zvonarev N., Golyandina N. Low-rank signal subspace: parameterization, projection and signal estimation // *Statistics and Its Interface*. — 2023. — Vol. 16, no. 1. — Pp. 117–132.
25. Golyandina N. Detection of signals by Monte Carlo singular spectrum analysis: multiple testing // *Statistics and Its Interface*. — 2023. — Vol. 16, no. 1. — Pp. 147–157.
26. Measuring gene expression noise in early Drosophila embryos: Nucleus-to-nucleus variability / N.E. Golyandina, D.M. Holloway, F.J.P. Lopes et al. // *Procedia Computer Science*. — 2012. — Vol. 9. — Pp. 373–382. — Proceedings of the International Conference on Computational Science, ICCS 2012.
27. Measuring gene expression noise in early Drosophila embryos: The highly dynamic compartmentalized micro-environment of the blastoderm is one of the main sources of noise / A.V. Spirov, N.E. Golyandina, D.M. Holloway et al. // *Lecture Notes in Computer Science (Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics)* / Ed. by M. Giacobini, L. Vanneschi, W. S. Bush. — Vol. 7246. — Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. — Pp. 177–188.

28. *Okhotnikov G., Golyandina N.* EOP time series prediction using singular spectrum analysis // CEUR Workshop Proceedings (MAChine Learning for EArth ObservatioN Workshop 2019) / Ed. by Th. Corpetti, D. Ienco, R. Interdonato et al. — 2019. — Vol. 2466.
29. *Golyandina N., Zhigljavsky A.* Singular Spectrum Analysis for Time Series. Springer Briefs in Statistics. — Springer, 2013.
30. *Golyandina N., Korobeynikov A., Zhigljavsky A.* Singular Spectrum Analysis with R. — Springer-Verlag Berlin Heidelberg, 2018.
31. *Golyandina N., Zhigljavsky A.* Singular Spectrum Analysis for Time Series. Springer Briefs in Statistics. — 2nd edition. — Springer, 2020.
32. Голяндина Н., Данилов Д. Обработка многомерных временных рядов с помощью метода “Гусеница” // Главные компоненты временных рядов / Под ред. Д. Данилов, А. Жиглявский. — 1997.
33. Голяндина Н. Э., Некруткин В. В., Степанов Д. В. Варианты метода «Гусеница»-SSA для анализа многомерных временных рядов // Труды II Международной конференции «Идентификация систем и задачи управления», SICPRO’03. Москва. — 2003. — С. 2139–2168.
34. Степанов Д., Голяндина Н. Варианты метода ‘Гусеница’-SSA для прогноза многомерных временных рядов // Сборник трудов IV Международной конференции по идентификации систем и проблемам управления (SICPRO 05). — Москва: Институт проблем управления им. В.А.Трапезникова, 2005. — С. 1831–1848.
35. *Golyandina N., Stepanov D.* SSA-based approaches to analysis and forecast of multidimensional time series // Proceedings of the 5th St.Petersburg Workshop on Simulation, June 26-July 2, 2005, St. Petersburg State University, St. Petersburg. — 2005. — Pp. 293–298.
36. *Alexandrov Th., Golyandina N.* Automatic extraction and forecast of time series cyclic components within the framework of SSA // Proceedings of the 5th St.Petersburg Workshop on Simulation. — St. Petersburg State University, 2005. — Pp. 45–50.
37. Александров Ф., Голяндина Н. Выбор параметров при автоматическом выделении трендовых и периодических составляющих временного ряда в рамках подхода “Гусеница”-SSA. // Труды IV Международной конференции “Идентификация систем и задачи управления” SICPR0’05. — 2005.
38. *Golyandina N., Vlassieva E.* First-order SSA-errors for long time series: model examples of simple noisy signals // Proceedings of the 6th St.Petersburg Workshop on Simulation Vol.1, June 28-July 4, 2009, St. Petersburg. — St.-Petersburg State University, 2009. — Pp. 314–319.
39. *Golyandina N., Usevich K.* An Algebraic View on Finite Rank in 2D-SSA // Proceedings of the 6th St.Petersburg Workshop on Simulation, June 28-July 4, St. Petersburg, Russia. — 2009. — Pp. 308–313.
40. *Golyandina N.E., Usevich K.D.* 2D-Extension of Singular Spectrum Analysis: Algorithm and Elements of Theory // Matrix Methods: Theory, Algorithms and Applications. — 2010. —

- Pp. 449–473.
41. Shlemov A., Golyandina N. Shaped extensions of Singular Spectrum Analysis // 21st International Symposium on Mathematical Theory of Networks and Systems, July 7–11, 2014. Groningen, The Netherlands. — 2014. — Pp. 1813–1820.
 42. Александров Ф., Голяндина Н. Автоматизация выделения трендовых и периодических составляющих временного ряда в рамках метода «Гусеница»-SSA // *Exponenta Pro (Математика в приложениях)*. — 2004. — № 3–4 (7–8). — С. 54–61.
 43. Golyandina N., Florinsky I., Usevich K. Filtering of Digital Terrain Models by 2D Singular Spectrum Analysis // *International Journal of Ecology & Development*. — 2007. — Vol. 8, no. F07. — Pp. 81–94.
 44. Голяндина Н.Э., Усевич К.Д., Флоринский И.В. Анализ сингулярного спектра для фильтрации цифровых моделей рельефа // *Геодезия и картография*. — 2008. — № 5. — С. 21–28.
 45. Alexandrov Th., Golyandina N., Spirov A. Singular Spectrum Analysis of Gene Expression Profiles of Early Drosophila embryo: Exponential-in-Distance Patterns // *Research Letters in Signal Processing*. — 2008. — Vol. 2008. — Pp. 1–5.
 46. Hall M. Jr. Combinatorial theory. — Wiley, New York, 1998.
 47. Linear recurring sequences over rings and modules / V. L. Kurakin, A. S. Kuzmin, A.V. Mikhalev, A. A. Nechaev // *Journal of Mathematical Sciences*. — 1995. — Vol. 76, no. 6. — Pp. 2793–2915.
 48. Bozzo E., Carniel R., Fasino D. Relationship between Singular Spectrum Analysis and Fourier analysis: Theory and application to the monitoring of volcanic activity // *Computers & Mathematics with Applications*. — 2010. — 08. — Vol. 60, no. 3. — Pp. 812–820.
 49. Harris T.J., Yan H. Filtering and frequency interpretations of Singular Spectrum Analysis // *Physica D*. — 2010. — Vol. 239. — Pp. 1958–1967.
 50. Kumar K. S., Rajesh R., Tiwari R. K. Regional and residual gravity anomaly separation using the singular spectrum analysis-based low pass filtering: a case study from Nagpur, Maharashtra, India // *Exploration Geophysics*. — 2018. — Vol. 49, no. 3. — Pp. 217–220.
 51. A simple SSA-based de-noising technique to remove ECG interference in EMG signals / J. Barrios-Muriel, F. Romero, F. J. Alonso, K. Gianikellis // *Biomedical Signal Processing and Control*. — 2016. — Vol. 30. — Pp. 117–126.
 52. Watson Ph. J. Identifying the Best Performing Time Series Analytics for Sea Level Research // Time Series Analysis and Forecasting: Selected Contributions from the ITISE Conference / Ed. by I. Rojas, H. Pomares. — Cham: Springer International Publishing, 2016. — Pp. 261–278.
 53. On time-variable seasonal signals: Comparison of SSA and Kalman filtering based approach / Q. Chen, M. Weigelt, N. Sneeuw, T. van Dam // VIII Hotine-Marussi Symposium on Mathematical Geodesy: Proceedings of the Symposium in Rome, 17–21 June, 2013 / Ed. by N. Sneeuw, P. Novák, M. Crespi, F. Sansò. — Cham: Springer International Publishing,

2016. — Pp. 75–80.
54. *Hyndman R. J., Athanasopoulos G.* Forecasting: Principles and Practice. — 2nd edition. — Australia: OTexts, 2018.
 55. Forecasting U.S. Tourist arrivals using optimal Singular Spectrum Analysis / H. Hassani, A. Webster, E. S. Silva, S. Heravi // *Tourism Management*. — 2015. — Vol. 46. — Pp. 322–335.
 56. *Vile J. L. nnd Gillard J. W., Harper P. R., Knight V. A.* Predicting Ambulance Demand Using Singular Spectrum Analysis // *Journal of the Operational Research Society*. — 2012. — Vol. 63, no. 11. — Pp. 1556–1565.
 57. *Iqelan B. M.* A Singular Spectrum Analysis Technique to Electricity Consumption Forecasting // *International Journal of Engineering Research and Applications*. — 2017. — Vol. 7, no. 3. — Pp. 92–100.
 58. *Hassani H., Heravi S., Zhigljavsky A.* Forecasting European industrial production with singular spectrum analysis // *International Journal of Forecasting*. — 2009. — Vol. 25, no. 1. — Pp. 103–118.
 59. *Hassani H., Heravi S., Zhigljavsky A.* Forecasting UK industrial production with multivariate singular spectrum analysis // *Journal of Forecasting*. — 2013. — Vol. 32, no. 5. — Pp. 395–408.
 60. *Broomhead D., King G.* Extracting qualitative dynamics from experimental data // *Physica D*. — 1986. — Vol. 20. — Pp. 217–236.
 61. *Fraedrich K.* Estimating dimensions of Weather and Climate Attractors // *Journal of the Atmospheric Sciences*. — 1986. — Vol. 43. — Pp. 419–432.
 62. *Colebrook J. M.* Continuous plankton records — zooplankton and environment, northeast Atlantic and North Sea, 1948–1975 // *Oceanologica Acta*. — 1978. — Vol. 1. — Pp. 9–23.
 63. *Vautard R., Ghil M.* Singular spectrum analysis in nonlinear dynamics, with applications to paleoclimatic time series // *Physica D*. — 1989. — Vol. 35. — Pp. 395–424.
 64. *Vautard R., Yiou P., Ghil M.* Singular-Spectrum Analysis: A toolkit for short, noisy chaotic signals // *Physica D*. — 1992. — Vol. 58. — Pp. 95–126.
 65. *Elsner J. B., Tsonis A. A.* Singular Spectrum Analysis: A New Tool in Time Series Analysis. — Plenum Press, 1996.
 66. Главные компоненты временных рядов: метод “Гусеница” / Под ред. Д. Данилов, А. Жиглявский. — СПбГУ, 1997.
 67. Факторный анализ в нефтяной геологии: Обзор / М. Д. Белонин, И. В. Татаринов, О. М. Калинин и др.; Под ред. А.М.Марголин. — Москва, ВИЕМС, 1971.
 68. *Белонин М. Д., А. Голубева В., Скублов Г. Т.* Факторный анализ в геологии. — Москва: Недра, 1982.
 69. *Buchstaber V.M.* Time series analysis and grassmannians // Applied Problems of Radon Transform / Ed. by S. Gindikin. — Providence: AMS, 1994. — Vol. 162 of *AMS Transactions – Series 2*. — Pp. 1–17.
 70. *Korobeynikov A., Shlemov A., Usevich K., Golyandina N.* — Rssa: A collection of methods for

- singular spectrum analysis, 2021. — R package version 1.0.4. URL: <http://CRAN.R-project.org/package=Rssa>.
71. *Yiou P., Sornette D., Ghil M.* Data-adaptive wavelets and multi-scale singular-spectrum analysis // *Physica D*. — 2000. — Vol. 142. — Pp. 254–290.
 72. *Sanei S., Hassani H.* Singular Spectrum Analysis of Biomedical Signals. — CRC press, 2015.
 73. *Hassani H., Thomakos D.* A review on Singular Spectrum Analysis for economic and financial time series // *Statistics and Its Interface*. — 2010. — Vol. 3, no. 3. — Pp. 377–397.
 74. *Trickett S.* F-xy eigenimage noise suppression // *Geophysics*. — 2003. — Vol. 68, no. 2. — Pp. 751–759.
 75. *Oropeza V.* The Singular Spectrum Analysis method and its application to seismic data denoising and reconstruction. — Master Thesis in University of Alberta. — 2010.
 76. *Basilevsky A., Hum Derek P. J.* Karhunen-Loéve Analysis of Historical Time Series With an Application to Plantation Births in Jamaica // *Journal of the American Statistical Association*. — 1979. — Vol. 74. — Pp. 284–290.
 77. Ефимов В., Галактионов Ю. О возможности прогнозирования циклических изменений численности млекопитающих // *Журнал общей биологии*. — 1983. — Т. 44, № 3. — С. 343–352.
 78. *Tufts D. W., Kumaresan R., Kirssteins I.* Data adaptive signal estimation by singular value decomposition of a data matrix // *Proc. IEEE*. — 1982. — Vol. 70, no. 6. — Pp. 684–685.
 79. *Ghil M., Vautard R.* Interdecadal oscillations and the warming trend in global temperature time series // *Nature*. — 1991. — Vol. 350. — Pp. 324–327.
 80. *Gantmacher F. R.* The Theory of Matrices. — Chelsea Publishing Company, New York 68, 1959.
 81. *de Prony G.* Essai expérimental et analytique sur les lois de la dilatabilité des fluides élastiques et sur celles de la force expansive de la vapeur de l'eau et la vapeur de l'alkool à différentes températures // *J de l'Ecole Polytechnique*. — 1795. — Vol. 1, no. 2. — Pp. 24–76.
 82. *Zhigljavsky A.* Singular spectrum analysis for time series: Introduction to this special issue // *Statistics and Its Interface*. — 2010. — Vol. 3, no. 3. — Pp. 255–258.
 83. Singular Spectrum Analysis for Effective Feature Extraction in Hyperspectral Imaging / J. Zabalza, J. Ren, Z. Wang et al. // *IEEE Geoscience and Remote Sensing Letters*. — 2014. — Vol. 11, no. 11. — Pp. 1886–1890.
 84. Novel two-dimensional singular spectrum analysis for effective feature extraction and data classification in hyperspectral imaging / J. Zabalza, J. Ren, J. Zheng et al. // *IEEE Geoscience and Remote Sensing Letters*. — 2015. — Vol. 53, no. 8. — Pp. 4418–4433.
 85. *von Buttlar J., Zscheischler J., Mahecha M. D.* An extended approach for spatiotemporal gapfilling: dealing with large and systematic gaps in geoscientific datasets // *Nonlinear Processes in Geophysics*. — 2014. — Vol. 21. — Pp. 203–215.
 86. *Groth A., Ghil M.* Monte Carlo Singular Spectrum Analysis (SSA) revisited: Detecting oscillator clusters in multivariate datasets // *Journal of Climate*. — 2015. — Vol. 28, no. 19.

- Pp. 7873–7893.
87. Tool wear estimation for different workpiece materials using the same monitoring system / D. R. Salgado, I. Cambero, J. M. H. Olivenza et al. // *Procedia Engineering*. — 2013. — Vol. 63. — Pp. 608–615.
 88. ECDSA Key Extraction from Mobile Devices via Nonintrusive Physical Side Channels / D. Genkin, L. Pachmanov, I. Pipman et al. // Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. — CCS '16. — New York, NY, USA: ACM, 2016. — Pp. 1626–1638.
 89. Economic Cycles and Their Synchronization: A Comparison of Cyclic Modes in Three European Countries / L. Sella, G. Vivaldo, A. Groth, M. Ghil // *Journal of Business Cycle Research*. — 2016. — Vol. 12, no. 1. — Pp. 25–48.
 90. Filho A. S. F., Lima G. A. R. Gap Filling of Precipitation Data by SSA - Singular Spectrum Analysis // *Journal of Physics: Conference Series*. — 2016. — Vol. 759, no. 1. — P. 012085.
 91. Trends in high sea levels of German North Sea gauges compared to regional mean sea level changes / C. Mudersbach, Th. Wahl, I. D. Haigh, J. Jensen // *Continental Shelf Research*. — 2013. — Vol. 65. — Pp. 111–120.
 92. Internal climate memory in observations and models / D. P. Monselesan, T. J. O’Kane, J. S. Risbey, J. Church // *Geophysical Research Letters*. — 2015. — Vol. 42, no. 4. — Pp. 1232–1242.
 93. Pepelyshev A., Zhigljavsky A. SSA analysis and forecasting of records for Earth temperature and ice extents // *Statistics and Its Interface*. — 2017. — Vol. 10, no. 1. — Pp. 151–163.
 94. Tampering Detection in Speech Signals by Semi-Fragile Watermarking Based on Singular-Spectrum Analysis / J. Karnjana, M. Unoki, P. Aimmanee, C. Wutiwiwatchai // Advances in Intelligent Information Hiding and Multimedia Signal Processing: Proceeding of the Twelfth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Nov., 21-23, 2016, Kaohsiung, Taiwan, Volume 1 / Ed. by J.-S. Pan, P.-W. Tsai, H.-C. Huang. — Cham: Springer International Publishing, 2017. — Pp. 131–140.
 95. Hudson I. L., Keatley M. R. Singular Spectrum Analytic (SSA) recombination and reconstruction of flowering: Signatures of climatic impacts // *Environmental Modeling & Assessment*. — 2017. — Feb. — Vol. 22, no. 1. — Pp. 37–52.
 96. Localizing microaneurysms in fundus images through Singular Spectrum Analysis / S. Wang, H. L. Tang, L. I. Al turk et al. // *IEEE Transactions on Biomedical Engineering*. — 2017. — May. — Vol. 64, no. 5. — Pp. 990–1002.
 97. Singular Spectrum Analysis and ARIMA Hybrid Model for Annual Runoff Forecasting / Q. Zhang, B.-D. Wang, B. He et al. // *Water Resources Management*. — 2011. — Vol. 25, no. 11. — Pp. 2683–2703.
 98. A failure time series prediction method based on UML model / W. Xin, L. Chao, X. Weiren, L. Ying // 2015 4th International Conference on Computer Science and Network Technology (ICCSNT). — Vol. 01. — 2015. — Pp. 62–70.

99. Lakshmi K., Rao A. R. M., Gopalakrishnan N. Singular spectrum analysis combined with ARMAX model for structural damage detection // *Structural Control and Health Monitoring*. — 2017. — Vol. 24, no. 9. — P. e1960. — STC-15-0200.R3.
100. Sivapragasam C., Liong S.-Y., Pasha M.F.K. Rainfall and runoff forecasting with SSA–SVM approach // *Journal of Hydroinformatics*. — 2001. — Vol. 3, no. 3. — Pp. 141–152.
101. Lisi F., Nicolis O., Sandri M. Combining Singular-Spectrum Analysis and neural networks for time series forecasting // *Neural Processing Letters*. — 1995. — Jul. — Vol. 2, no. 4. — Pp. 6–10.
102. A hybrid model based on Singular Spectrum Analysis and Support Vector Machines Regression for failure time series prediction / X. Wang, J. Wu, C. Liu et al. // *Quality and Reliability Engineering International*. — 2016. — Vol. 32, no. 8. — Pp. 2717–2738. — QRE-16-0186.R2.
103. A neuro-fuzzy combination model based on singular spectrum analysis for air transport demand forecasting / Y. Xiao, J. J. Liu, Y. Hu et al. // *Journal of Air Transport Management*. — 2014. — Vol. 39. — Pp. 1–11.
104. Wu C. L., Chau K. W. Rainfall–runoff modeling using artificial neural network coupled with singular spectrum analysis // *Journal of Hydrology*. — 2011. — Vol. 399, no. 3. — Pp. 394–409.
105. Unit operational pattern analysis and forecasting using EMD and SSA for industrial systems / Z. Yang, C. Bingham, W.-K. Ling et al. // Advances in Intelligent Data Analysis XI: 11th International Symposium, IDA 2012, Helsinki, Finland, October 25–27, 2012. Proceedings / Ed. by J. Hollmén, F. Klawonn, A. Tucker. — Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. — Pp. 416–423.
106. Andrew A. L. Eigenvectors of certain matrices // *Linear Algebra and its Applications*. — 1973. — Vol. 7, no. 2. — Pp. 151–162.
107. Usevich K. On signal and extraneous roots in Singular Spectrum Analysis // *Statistics and Its Interface*. — 2010. — Vol. 3, no. 3. — Pp. 281–295.
108. Roy R., Kailath T. ESPRIT: Estimation of Signal Parameters via Rotational Invariance Techniques // *Acoustics, Speech and Signal Processing, IEEE Transactions on*. — 1989. — Vol. 37. — Pp. 984–995.
109. Barkhuijsen H., de Beer R., van Ormondt D. Improved algorithm for noniterative time-domain model fitting to exponentially damped magnetic resonance signals // *Journal of Magnetic Resonance*. — 1987. — Vol. 73. — Pp. 553–557.
110. Algorithm for time-domain NMR data fitting based on total least squares / S. Van Huffel, H. Chen, C. Decanniere, P. van Hecke // *Journal of Magnetic Resonance, Series A*. — 1994. — Vol. 110. — Pp. 228–237.
111. Kumaresan R., Tufts D. W. Estimating the Angles of Arrival of Multiple Plane Waves // *IEEE Transactions on Aerospace and Electronic Systems*. — 1983. — Vol. AES-19, no. 1. — Pp. 134–139.

112. *Pakula L.* Asymptotic zero distribution of orthogonal polynomials in sinusoidal frequency estimation // *IEEE Transactions on Information Theory*. — 1987. — Vol. 33, no. 4. — Pp. 569–576.
113. *Korobeynikov A.* Computation- and Space-Efficient Implementation of SSA // *Statistics and Its Interface*. — 2010. — Vol. 3, no. 3. — Pp. 357–368.
114. *Abraham B., Ledolter J.* Statistical Methods for Forecasting. — Toronto: Wiley, 1983.
115. *Keeling C. D., Whorf T. P.* Atmospheric CO₂ concentrations — Mauna Loa Observatory, Hawaii, 1959–1997. — Scripps Institution of Oceanography (SIO), University of California, La Jolla, California USA 92093-0220. — 1997.
116. *Van Loan Ch. F.* Generalizing the Singular Value Decomposition // *SIAM Journal on Numerical Analysis*. — 1976. — Vol. 13. — Pp. 76–83.
117. *De Moor B., Golub G.* The Restricted Singular Value Decomposition: Properties and applications // *SIAM Journal on Matrix Analysis and Applications*. — 1991. — Vol. 12, no. 3. — Pp. 401–425.
118. *Paige C., Saunders M.* Towards a generalized singular value decomposition // *SIAM Journal on Numerical Analysis*. — 1981. — Vol. 18. — Pp. 398–405.
119. *Andrews D.F., Herzberg A.M.* Data. A Collection of Problems from Many Fields for the Student and Research Worker. — New York: Springer, 1985.
120. *Hyndman R., Yang Y.* TSDL: Time Series Data Library. — 2018. — URL: <https://github.com/FinYang/tSDL> (online; accessed: 10/02/2023).
121. *Hyvärinen A.* Fast and robust fixed-point algorithms for independent component analysis // *IEEE Transactions on Neural Networks*. — 1999. — may. — Vol. 10, no. 3. — Pp. 626–634.
122. *Hyvärinen A., Karhunen J., Oja E.* Independent component analysis. Adaptive and learning systems for signal processing, communications, and control. — New York: Wiley, 2001.
123. *Huber P. J.* Projection Pursuit // *Annals of Statistics*. — 1985. — Vol. 13, no. 2. — Pp. 435–475.
124. *Hyvärinen A., Oja E.* Independent component analysis: algorithms and applications // *Neural Networks*. — 2000. — Vol. 13, no. 4-5. — Pp. 411 – 430.
125. *Marchini J. L., Heaton C., Riple B. D.* — FASTICA: FastICA Algorithms to Perform ICA and Projection Pursuit, 2021. — URL: <http://CRAN.R-project.org/package=fastICA>.
126. Blind Source Separation of Cardiac Murmurs from Heart Recordings / A. Pietilä, M. El-Segaeier, R. Vigário, E. Pesonen // Independent Component Analysis and Blind Signal Separation / Ed. by J. Rosca, et al. — Springer Berlin / Heidelberg, 2006. — Vol. 3889 of *Lecture Notes in Computer Science*. — Pp. 470–477.
127. Indeterminacy and identifiability of blind identification / L. Tong, R.-W. Liu, V.C. Soon, Y.-F. Huang // *IEEE Transactions on Circuits and Systems*. — 1991. — may. — Vol. 38, no. 5. — Pp. 499–509.
128. A blind source separation technique using second-order statistics / A. Belouchrani, K. Abed-Meraim, J.-F. Cardoso, E. Moulines // *IEEE Transactions on Signal Processing*. — 1997. —

- Vol. 45, no. 2. — Pp. 434–444.
129. *Golyandina N., Zhornikova P.* On automated identification in singular spectrum analysis for different types of objects. — 2023. — URL: <https://arxiv.org/abs/2302.08993>.
 130. *Александров Ф. И.* Разработка программного комплекса автоматического выделения и прогноза аддитивных компонент временных рядов в рамках подхода «Гусеница»-SSA: диссертация на соискание степени к.ф.-м.н. / Санкт-петербург. гос. ун-т. — Санкт-Петербург, 2006. — 152 с.
 131. *Fisher R. A.* Tests of significance in Harmonic Analysis // *Proceedings of the Royal Society A.* — 1929. — Vol. 125. — Pp. 54–59.
 132. *Schoellhamer D.H.* Singular spectrum analysis for time series with missing data // *Geophysical Research Letters.* — 2001. — Vol. 28, no. 16. — Pp. 3187–3190.
 133. *Kondrashov D., Ghil M.* Spatio-temporal filling of missing points in geophysical data sets // *Nonlinear Processes in Geophysics.* — 2006. — Vol. 13, no. 2. — Pp. 151–159.
 134. *Beckers J., Rixen M.* EOF calculations and data filling from incomplete oceanographic data sets // *Journal of Atmospheric and Oceanic Technology.* — 2003. — Vol. 20. — Pp. 1839–1856.
 135. *Brown R. G.* Smoothing, forecasting and prediction of discrete time series. — Prentice-Hall Englewood Cliffs, N.J, 1963.
 136. *Chu M. T., Funderlic R. E., Plemmons R. J.* Structured low rank approximation // *Linear Algebra and its Applications.* — 2003. — Vol. 366. — Pp. 157–172.
 137. Exact and Approximate Modeling of Linear Systems / I. Markovsky, J. Willems, S. Van Huffel, B. De Moor. — SIAM, 2006.
 138. *Markovsky I., Usevich K.* Software for Weighted Structured Low-Rank Approximation // *Computational and Applied Mathematics.* — 2014. — Vol. 256. — Pp. 278–292.
 139. *Gillard J., Zhigljavsky A.* Optimization challenges in the structured low rank approximation problem // *Journal of Global Optimization.* — 2013. — Vol. 57, no. 3. — Pp. 733–751.
 140. *Zvonarev N. K.* Search for Weights in the Problem of Finite-Rank Signal Estimation in the Presence of Random Noise // *Vestnik St. Petersburg University, Mathematics.* — 2021. — jul. — Vol. 54, no. 3. — Pp. 236–244.
 141. *Kundur D., Hatzinakos D.* Blind image deconvolution // *IEEE Signal Processing Magazine.* — 1996. — May. — Vol. 13, no. 3. — Pp. 43–64.
 142. *Michailovich O., Tannenbaum A.* Blind deconvolution of medical ultrasound images: A parametric inverse filtering approach // *IEEE Transactions on Image Processing.* — 2007. — Dec. — Vol. 16, no. 12. — Pp. 3005–3019.
 143. *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, no. 1. — Pp. 70–88.
 144. *Allen G. I., Grosenick L., Taylor J.* A generalized least-square matrix decomposition // *Journal of the American Statistical Association.* — 2014. — Vol. 109, no. 505. — Pp. 145–159.
 145. *Shaman P.* An Approximate Inverse for the Covariance Matrix of Moving Average and

- Autoregressive Processes // *Annals of Statistics*. — 1975. — 03. — Vol. 3, no. 2. — Pp. 532–538.
146. *Siddiqui M. M.* On the Inversion of the Sample Covariance Matrix in a Stationary Autoregressive Process // *The Annals of Mathematical Statistics*. — 1958. — 06. — Vol. 29, no. 2. — Pp. 585–588.
 147. *Jury E. I.* Theory and application of the Z-transform method. — New York, NY: Wiley, 1964.
 148. *Vaughan, S.* A simple test for periodic signals in red noise // *Astronomy & Astrophysics*. — 2005. — Vol. 431, no. 1. — Pp. 391–403.
 149. *Louca S., Doebeli M.* Detecting cyclicity in ecological time series // *Ecology*. — 2015. — Vol. 96, no. 6. — Pp. 1724–1732.
 150. *Grenander U., Szegö G.* Toeplitz Forms and their Applications. — New York: Chelsea, 1984.
 151. *Allen M. R., Smith L. A.* Monte Carlo SSA: Detecting irregular oscillations in the Presence of Colored Noise // *Journal of Climate*. — 1996. — Vol. 9, no. 12. — Pp. 3373–3404.
 152. *Allen M. R., Robertson A. W.* Distinguishing modulated oscillations from coloured noise in multivariate datasets // *Climate Dynamics*. — 1996. — Vol. 12, no. 11. — Pp. 775–784.
 153. *Garnot V. S. F., Groth A., Ghil M.* Coupled Climate-Economic Modes in the Sahel's Interannual Variability // *Ecological Economics*. — 2018. — Vol. 153. — Pp. 111 – 123.
 154. Evidence of deterministic components in the apparent randomness of GRBs: clues of a chaotic dynamic. / G. Greco, R. Rosa, G. Beskin et al. // *Scientific reports*. — 2011. — Vol. 1. — P. 91.
 155. *Jemwa G. T., Aldrich C.* Classification of process dynamics with Monte Carlo singular spectrum analysis // *Computers & Chemical Engineering*. — 2006. — Vol. 30, no. 5. — Pp. 816 – 831.
 156. *Palus M., Novotná D.* Detecting modes with nontrivial dynamics embedded in colored noise: enhanced Monte Carlo SSA and the case of climate oscillations // *Physics Letters A*. — 1998. — Vol. 248, no. 2. — Pp. 191 – 202.
 157. *Palus M., Novotná D.* Enhanced Monte Carlo Singular System Analysis and detection of period 7.8 years oscillatory modes in the monthly NAO index and temperature records // *Nonlinear Processes in Geophysics*. — 2004. — Vol. 11, no. 5/6. — Pp. 721–729.
 158. *Trench W. F.* Spectral decomposition of Kac-Murdock-Szegő matrices. — Unpublished paper from the Selected works of William F. Trench. URL: https://works.bepress.com/william_trench/133/ (online; accessed: 10/02/2023).
 159. Surrogate data for hypothesis testing of physical systems / G. Lancaster, D. Iatsenko, A. Pidde et al. // *Physics Reports*. — 2018. — Vol. 748. — Pp. 1–60.
 160. *Бояров А.* — Изучение статистических свойств Monte Carlo SSA. — Master's thesis, С.-Петербургский государственный университет, 2012.
 161. *Golyandina N.* R-scripts for multiple testing in Monte Carlo singular spectrum analysis. — <https://zenodo.org/record/4568495>. — 2021.

162. Nekrutkin V. Perturbation expansions of signal subspaces for long signals // *Statistics and Its Interface*. — 2010. — Vol. 3. — Pp. 297–319.
163. Björck Å. Numerical Methods for Least Squares Problems. — SIAM, 1996.
164. Badeau R., Richard G., David B. Performance of ESPRIT for Estimating Mixtures of Complex Exponentials Modulated by Polynomials // *IEEE Transactions on Signal Processing*. — 2008. — Vol. 56, no. 2. — Pp. 492–504.
165. Djermoune E.-H., Tomczak M. Perturbation Analysis of Subspace-Based Methods in Estimating a Damped Complex Exponential // *IEEE Transactions on Signal Processing*. — 2009. — Vol. 57, no. 11. — Pp. 4558–4563.
166. Rife D., Boorstyn R. Single tone parameter estimation from discrete-time observations // *IEEE Trans Inf Theory*. — 1974. — Vol. 20, no. 5. — Pp. 591–598.
167. Stoica P., Jakobsson A., L. Jian. Cisoid parameter estimation in the colored noise case: asymptotic Cramer-Rao bound, maximum likelihood, and nonlinear least-squares // *IEEE Transactions on Signal Processing*. — 1997. — Vol. 45, no. 8. — Pp. 2048–2059.
168. 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. — Pp. 1–41.
169. Weare Bryan C., Nasstrom John S. Examples of Extended Empirical Orthogonal Function Analyses // *Monthly Weather Review*. — 1982. — Vol. 110, no. 6. — Pp. 481–485.
170. Broomhead D.S., King G.P. On the qualitative analysis of experimental dynamical systems // Nonlinear Phenomena and Chaos / Ed. by S. Sarkar. — Bristol: Adam Hilger, 1986. — Pp. 113–144.
171. Hannachi A., Jolliffe I. T., Stephenson D. B. Empirical Orthogonal Functions and Related Techniques in Atmospheric Science: A Review // *International Journal of Climatology*. — 2007. — Vol. 27, no. 9. — Pp. 1119–1152.
172. Hassani H., Mahmoudvand R. Multivariate singular spectrum analysis: a general view and vector forecasting approach // *International Journal of Energy and Statistics*. — 2013. — Vol. 01, no. 01. — Pp. 55–83.
173. Golub G. H., Van Loan Ch. F. Matrix computations (3rd ed.). — Baltimore, MD, USA: Johns Hopkins University Press, 1996.
174. Mourrain B., Pan V. Y. Multivariate polynomials, duality, and structured matrices // *Journal of Complexity*. — 2000. — Vol. 16, no. 1. — Pp. 110–180.
175. Heinig G. Generalized inverses of Hankel and Toeplitz mosaic matrices // *Linear Algebra and its Applications*. — 1995. — Vol. 216, no. 0. — Pp. 43–59.
176. Rodríguez-Aragón L. J., Zhigljavsky A. Singular Spectrum Analysis for Image Processing // *Statistics and Its Interface*. — 2010. — Vol. 3, no. 3. — Pp. 419–426.
177. Ade F. Characterization of Textures by 'Eigenfilters' // *Signal Processing*. — 1983. — Vol. 5, no. 5. — Pp. 451–457.
178. Monadjemi A. Towards Efficient Texture Classification and Abnormality Detection: Ph.D. thesis / University of Bristol. — 2004.

179. *Trickett S.* F-xy Cadzow Noise Suppression // 78th Annual International Meeting, SEG, Expanded Abstracts. — 2008. — Pp. 2586–2590.
180. *Rouquette S, Najim M.* Estimation of Frequencies and Damping Factors by Two-Dimensional ESPRIT Type Methods // *IEEE Transactions on Signal Processing*. — 2001. — Vol. 49, no. 1. — Pp. 237–245.
181. ESPIRiT — an Eigenvalue Approach to Autocalibrating Parallel MRI: Where SENSE Meets GRAPPA / M. Uecker, P. Lai, M. J. Murphy et al. // *Magnetic Resonance in Medicine*. — 2014. — Vol. 71, no. 3. — Pp. 990–1001.
182. *Korobeynikov A., Larsen R. M., Wu K. J., Yamazaki I.* — SVD: Interfaces to various state-of-art SVD and eigensolvers, 2023. — URL: <http://CRAN.R-project.org/package=svd>.
183. Achievements of the Earth orientation parameters prediction comparison campaign / M. Kalarus, H. Schuh, W. Kosek et al. // *Journal of Geodesy*. — 2010. — Vol. 84. — Pp. 587–596.
184. *Malkin Z.* Employing combination procedures to short-time EOP prediction // *Artificial Satellites*. — 2010. — Vol. 45, no. 2. — Pp. 87–93.
185. *Kosek W.* Future improvements in EOP prediction // *Geodesy for Planet Earth*. — Springer, 2012. — Vol. 136. — Pp. 513–520.
186. *Malkin Z.* Free core nutation and geomagnetic jerks // *Journal of Geodynamics*. — 2013. — Vol. 72. — Pp. 53–58.
187. *Миллер Н.* Прогнозирование движения полюса с помощью ССА // Известия ГАО в Пулково. — 2016. — Т. 223. — С. 119–123.
188. Polar motion prediction using the combination of SSA and Copula-based analysis / S. Modiri, S. Belda, R. Heinkelmann et al. // *Earth, Planets and Space*. — 2018. — Vol. 70, no. 1. — P. 115.
189. International Earth rotation and reference systems service Earth orientation parameters EOP (IERS) 14 C04. — URL: https://hpiers.obspm.fr/iers/eop/eopc04/eopc04_IAU2000_62-now.
190. Pulkovo EOP and Reference Systems Analysis Center (PERSAC). — URL: <http://www.gao.spb.ru/english/as/persac/eopc04/>.
191. BULLETIN A — all available versions. — URL: <https://www.iers.org/IERS/EN/Publications/Bulletins/bulletins.html>.
192. *Silverman B. W.* Density estimation for statistics and data analysis. — Routledge, 1998.
193. *Silverman B. W.* Using kernel density estimates to investigate multimodality // *Journal of the Royal Statistical Society: Series B (Methodological)*. — 1981. — sep. — Vol. 43, no. 1. — Pp. 97–99.
194. *Jones M. C., Marron J. S., Sheather S. J.* A brief survey of bandwidth selection for density estimation // *Journal of the American Statistical Association*. — 1996. — Vol. 91, no. 433. — Pp. 401–407.
195. *Good I. J., Gaskins R. A.* Density estimation and bump-hunting by the penalized likelihood

- method exemplified by scattering and meteorite data // *Journal of the American Statistical Association*. — 1980. — Vol. 75, no. 369. — Pp. 42–56.
196. *Lambert Ph., Eilers P.H.C.* Bayesian density estimation from grouped continuous data // *Computational Statistics & Data Analysis*. — 2009. — Vol. 53, no. 4. — Pp. 1388–1399.
 197. Generation and Interpretation of Morphogen Gradients: A Subject Collection from Cold Spring Harbor Perspectives in Biology / Ed. by J. Briscoe, P. A. Lawrence, J.-P. Vincent. — Cold Spring Harbor, N.Y.: Cold Spring Harbor Laboratory Press, 2010.
 198. *Houchmandzadeh B, Wieschaus E, Leibler S.* Establishment of developmental precision and proportions in the early Drosophila embryo // *Nature*. — 2002. — Vol. 415. — Pp. 798–802.
 199. *Alonso F.J., Castillo J.M.Del, Pintado P.* Application of singular spectrum analysis to the smoothing of raw kinematic signals // *Journal of Biomechanics*. — 2005. — Vol. 38, no. 5. — Pp. 1085–1092.
 200. Formation of the bicoid morphogen gradient: an mRNA gradient dictates the protein gradient / A.V. Spirov, K. Fahmy, M. Schneider et al. // *Development*. — 2009. — Vol. 136. — Pp. 605–614.
 201. Removal of background signal from *in situ* data on the expression of segmentation genes in Drosophila / E. Myasnikova, M. Samsonova, D. Kosman, J. Reinitz // *Development Genes and Evolution*. — 2005. — Vol. 215. — Pp. 320–326.
 202. Analysis of pattern precision shows that Drosophila segmentation develops substantial independence from gradients of maternal gene products / D.M. Holloway, L.G. Harrison, D. Kosman et al. // *Developmental Dynamics*. — 2006. — Vol. 235. — Pp. 2949–2960.
 203. Characterization of the *Drosophila* segment determination morphome / S. Surkova, D. Kosman, K. Kozlov et al. // *Developmental Biology*. — 2008. — Vol. 313, no. 2. — Pp. 844–862.
 204. Calibrationless parallel imaging reconstruction based on structured low-rank matrix completion / P. J. Shin, P. E. Z. Larson, M. A. Ohliger et al. // *Magnetic Resonance in Medicine*. — 2014. — Vol. 72, no. 4. — Pp. 959–970.
 205. A quantitative spatiotemporal atlas of gene expression in the *Drosophila* blastoderm / Ch. Fowlkes, C. L. L. Hendriks, S. V. E. Keränen et al. // *Cell*. — 2008. — Vol. 133, no. 2. — Pp. 364–374.
 206. A digital framework to build and visualize and analyze a gene expression atlas with cellular resolution in zebrafish early embryogenesis / C. Castro-González, M.A. Luengo-Oroz, L. Duloquin et al. // *PLoS Computational Biology*. — 2014. — Vol. 10, no. 6. — P. e1003670.
 207. Automatic channel unmixing for high-throughput quantitative analysis of fluorescence images / C. L. Luengo Hendriks, S. V. Keränen, M. D. Biggin, D. W. Knowles // *Optics express*. — 2007. — Vol. 15, no. 19. — Pp. 12306–12317.
 208. *Shlemov A., Golyandina N., Korobeynikov A. et al.* BioSSA. — 2014. — URL: <http://biossa.github.io/>.
 209. *Snyder J. P.* Map projections: A working manual. Geological Survey Bulletin Series. —

- Geological Survey (U.S.), 1987.
- 210. *Anjyo K., Lewis J. P., Pighin F.* Scattered Data Interpolation for Computer Graphics // ACM SIGGRAPH 2014 Courses. — SIGGRAPH '14. — New York, NY, USA: ACM, 2014. — Pp. 27:1–27:69.
 - 211. *Pion S., Teillaud M.* 3D Triangulations // CGAL User and Reference Manual. — CGAL Editorial Board, 2014.
 - 212. *Park R. E.* Estimation with Heteroscedastic Error Terms // *Econometrica*. — 1966. — Vol. 34, no. 4. — P. 888.