

Тензорная декомпозиция и прогноз для набора временных рядов

Сёмкин Кирилл¹

¹semkin.ki@phystech.edu

Вадим Стрижов²

²vadim.swifton@gmail.com

Аннотация

Декомпозиция временного ряда часто применяется для получения его структуры: разложения на простые и/или интерпретируемые составляющие, выявление периодичности, избавление от шума и т.д. В случае же набора нескольких рядов игнорирование взаимосвязей между ними может приводить к некачественным и ложным разложениям и предсказаниям. В данной работе предлагается метод декомпозиции, учитывающий фактор связанности и основанный на классическом методе SSA (Гусеница) и тензорных разложениях, а также способ прогноза будущих значений ряда. Предложенный подход сравнивается с похожим mSSA, основанным на матричных разложениях, в математических свойствах и при обработке синтетических и реальных данных (потребление электроэнергии, акселерометрия).

Ключевые слова: декомпозиция, прогноз, SVD, SSA, тензорные разложения.

Введение

Настоящая работа посвящена методам декомпозиции и построения прогноза для *набора временных рядов* - нескольких рассматриваемых вместе сигналов $\{x_i(t)\}_{i=1}^m$, где $t \in \overline{1, \vec{N}}$.

- **Аддитивная декомпозиция** временного ряда $x(t)$ - представление его в виде суммы некоторых сигналов $\{c_i(t)\}$:

$$x(t) = c_1(t) + \dots + c_n(t) = \sum_{i=1}^n c_i(t)$$

- **Прогноз** для ряда $x(t), t \in \overline{1, \vec{N}}$ с *горизонтом прогнозирования* k - набор предлагаемых будущих значений ряда $\hat{x}(t), t \in \overline{N+1, N+k}$ в соответствии с некоторым принципом (для нас это будет вводимая ниже модель рядов).

Методы решения задачи декомпозиции [4, 3, 2] и прогноза [8, 4, 14] для единственно рассматриваемого сигнала не могут быть просто перенесены на наборы временных рядов в случае их значимой взаимной зависимостью друг от друга. Для иллюстрации достаточно взять модель хищник-жертва [17]: размер популяции одних особей зависит от размера популяции других и наоборот. Везде далее мы будем предполагать некую связь рядов, строгое определение которой будет раскрыто при определении рассматриваемых далее моделей. Выявление же этой связи — отдельная задача, см., например, [5, 11].

Рассмотрим несколько подходов к рассматриваемому вопросу. *Векторная авторегрессия* (VAR) [1, 15] представляет собой линейную стохастическую модель динамики набора временных рядов. Если из значений сигналов в момент t составить вектор $\mathbf{x}_t = (x_1(t) \dots x_m(t))^T$, то динамика имеет вид:

$$\mathbf{x}_t = \boldsymbol{\mu} + \sum_{i=1}^p A_i \mathbf{x}_{t-i} + \mathbf{u}_t$$

Здесь $\boldsymbol{\mu}$ - некоторый константный вектор, A_i матрицы $m \times m$, \mathbf{u}_t - шум (например, реализации белого шума $WN(t)$). Как мы видим, связь рядов задаётся линейными преобразованиями A_i , и каждый сигнал зависит как от собственных значений в прошлом, так и от прошлого других сигналов *линейным образом*.

Модель *рекуррентных нейронных сетей* (RNN) [7, 13] похожа на предыдущую, но связывает ряды с прошлыми значениями уже более сложным, нелинейным образом (см. рис.1). На каждом шаге происходит вычисление скрытого представления h_t с помощью него на предыдущем шаге h_{t-1} и текущих значений рядов \mathbf{x}_t , см. рис.2. Оно инкапсулирует в себе всю информацию о прошлом и используется для прогноза [16, 6].

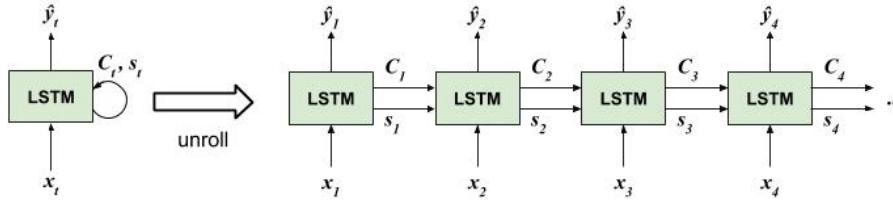


Рис. 1: Принцип работы RNN-сети для временного ряда

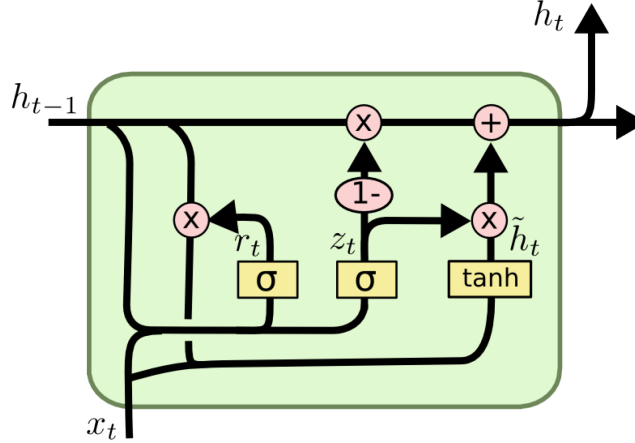


Рис. 2: Вариант строения RNN-ячейки

В данной работе предлагается использовать подход динамических систем: за основу взята модель *собственного пространства сигнала* и идея метода SSA [10]. Введём необходимые определения.

- **k -ый вектор задержки** $\overleftarrow{x}_k = (x_{k-L+1} \dots x_k)^T$, размер всех таких векторов L , также называемый **длина окна**.
- **Траекторная матрица** сигнала $x(t)$ есть $T = [\overleftarrow{x}_1 \dots \overleftarrow{x}_{N-L+1}]$
- **Траекторный тензор** набора сигналов $\{x_i(t)\}_{i=1}^m$ есть \mathbf{T} - 3ёх индексный тензор, состоящий из матриц задержек T_i всех сигналов, состыкованных по третьему измерению. Если же все T_i поставить друг за другом и составить одну большую матрицу, то будем обозначать её \mathbf{T}_m

Пусть существует динамическая система, траектории $\mathbf{y}(t)$ которой лежат на некотором многообразии M пространства состояний. Возьмём пока единственный наблюдаемый

сигнал $x(t)$ и предположим, что он есть результат отображения в \mathbb{R} некоторой конкретной траектории. Опираясь на теорему Такенса [12], аппроксимируем её векторами задержек $\{\vec{x}_k\}_{k=1}^{N-l+1}$ (на самом деле построим диффеоморфизм между M и \mathbb{R}^L). Найдя "удобный" базис в $Lin(\{\vec{x}_k\})$, мы получим полное описание M в терминах диффеоморфного ему \mathbb{R}^L , что и будет являться собственным пространством для $x(t)$.

Для этого применим SVD-разложение к траекторной матрице $T = \sum \sigma_i \mathbf{u}_i \mathbf{v}_i$, тогда $\{\mathbf{u}_i\}_{i=1}^r$ - искомый ортонормированный базис. С помощью него строится прогноз будущих значений сигнала, а на основании SVD производится построение декомпозиции ряда простым формированием групп из факторов $\{\sigma_i \mathbf{u}_i \mathbf{v}_i\}$. Подробное описание этого будет дано ниже. По-сути, сейчас был введён метод SSA.

Для набора временных рядов используется аналогичный подход, только теперь производится *каноническое* (CPD) разложение траекторного тензора \mathbf{T} , или то же SVD для \mathbf{T}_m . В зависимости от выбранного объекта методы будут называться **tSSA** и **mSSA** соответственно. Способ прогнозирования и декомпозиции концептуально ничем не отличается от методологии SSA.

В отличие от VAR и RNN, наша модель имеет мотивированную и чёткую структуру, основанную на динамических системах. Это значит, что многие физические, экономические и др. системы, порождающие сигналы во времени, могут быть описаны и обработаны нашими методами. Также, как будет показано далее, для mSSA и tSSA нет нужды в процедурах обучения/оптимизации, все необходимые вычисления — это разложения \mathbf{T} или \mathbf{T}_m . Наконец, из описания VAR и RNN вообще говоря не понятно, как построить декомпозицию рядов.

Далее будет дано математическое описание алгоритмов mSSA и tSSA, их особенности и сравнение в подходах к декомпозиции многомерных рядов. Будет описан метод построения прогноза для mSSA и tSSA, основанный на способе прогнозирования в классическом SSA. После данные методы тестируются на разложении и предсказании набора временных рядов: данные потребления и выработки электроэнергии, показания акселерометра движущегося пешехода. Анализируется ошибка, возникающая при разложении рядов, интерпретируемость разложения, качество предсказания на отложенных значениях выборки. Также будет эмпирически продемонстрировано некоторое важное свойство прогноза в рассматриваемых методах, см. .

Теоретическая часть

Пусть имеем набор временных рядов $\{x_i(t_j)\}_{i=1}^m$, где сетка по времени $t_j \in \overrightarrow{1, N}$. Стоит задача разложения временных рядов на аддитивные компоненты:

$$\begin{aligned} x_1(t) &= f_1(t) + f_2(t) + \dots + f_{n_1}(t) \\ x_2(t) &= g_1(t) + g_2(t) + \dots + g_{n_2}(t) \\ &\dots \end{aligned}$$

Формулировка задачи слишком общая, ведь таких разложений для произвольной функции можно построить сколь угодно много и каким угодно способом: можно раскладывать по индикаторным функциям (любую измеримую функцию можно так разложить), в ряды Фурье и т.д. Мы же будем полагаться на свойства вводимых далее методов обработки временных рядов для поиска этого разложения, а также для построения прогноза.

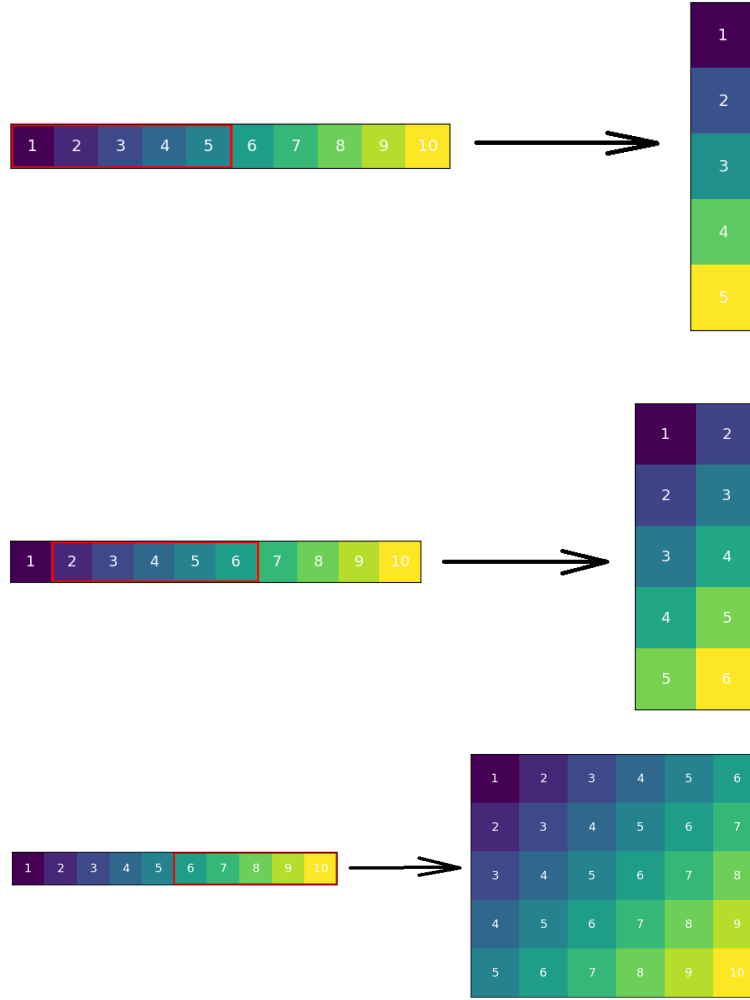


Рис. 3: Построение ганкелевой матрицы

Метод SSA

Пусть имеем пока один временной ряд. Подход "Гусеницы" предполагает наличие некоторой динамической системы, порождающей наблюдаемый ряд, и опирается на теорему Такенса об аппроксимации многообразия, в котором лежат траектории системы, так называемыми *векторами временных задержек* ([12]). Для построения этих векторов выбирается длина окна L , далее это окно "скользит" по временному ряду, выделяя в нём L значений (см. рис.3). Формально k -ый вектор задержек есть $\mathbf{x}_k = (x(t_k) \ x(t_{k+1}) \ \dots \ x(t_{k+L-1}))$. Далее данные вектора собираются в матрицу T_{ij} по столбцам, у которой в результате все антидиагональные элементы $i + j = \text{const}$ равны. Такие матрицы называются *ганкелевыми*, а в терминах временных рядов матрица T называется *траекторной* или *матрицей задержек*.

Далее к матрице применяется SVD-разложение $T = \sum \sigma_i \mathbf{u}_i \mathbf{v}_i$ (знак транспонирования для \mathbf{v}_i опускается), получая таким образом тот самый адаптивный ортонормированный базис $\{\mathbf{u}_i\}_{i=1}^r$ в пространстве векторов задержек. Компоненты с малыми σ_i интерпретируются как шумовые.

Для декомпозиции исходного ряда предлагается следующее: пусть $x(t) = f_1(t) + f_2(t)$ и пусть мы построили ганкелевы матрицы X_1, X_2 для $f_1(t)$ и $f_2(t)$, тогда $T = X_1 + X_2$. Как известно, SVD-разложение матриц единственно (разложение по ортонормированному базису), т.е. для матриц X_1 и X_2 это разложение по тем же базисам $\{\mathbf{u}_i\}_{i=1}^r, \{\mathbf{v}_i\}_{i=1}^r$. Главное предположение здесь — возможность разделить эти базисы на две непересекающиеся

группы: $(\{\mathbf{u}_i^1\}_{i=1}^{r_1}, \{\mathbf{v}_i^1\}_{i=1}^{r_1})$ и $(\{\mathbf{u}_i^2\}_{i=1}^{r_2}, \{\mathbf{v}_i^2\}_{i=1}^{r_2})$, где $r_1 + r_2 = r$. Тогда возможно выделить декомпозицию исходного ряда из его сингулярного разложения:

$$T = \sum_i^r \sigma_i \mathbf{u}_i \mathbf{v}_i = \sum_i^{r_1} \sigma_i^1 \mathbf{u}_i^1 \mathbf{v}_i^1 + \sum_i^{r_2} \sigma_i^2 \mathbf{u}_i^2 \mathbf{v}_i^2 = X_1 + X_2$$

Получив X_1, X_2 можно однозначно восстановить $\{f_1(t_j)\}, \{f_2(t_j)\}$, для этого достаточно взять первый столбец и последнюю строки в этих матрицах и соединить в один вектор друг за другом — это просто следует из способа построения матриц векторов задержек и их ганкелевой структуры. Таким же образом можно разделять спектральное разложение на большее число групп и по той же процедуре извлекать составляющие сигнала, если гипотеза возможности такого разделения верна.

Вводимое предположение имеет следующий смысл: т.к. вектора задержек для $f_1(t)$ и $f_2(t)$ лежат в ортогональных друг другу пространствах, порождённых $\{\mathbf{u}_i^1\}_{i=1}^{r_1}$ и $\{\mathbf{u}_i^2\}_{i=1}^{r_2}$, то и исходные вектора задержек двух рядов ортогональны друг другу:

$$f_1(t_i)f_2(t_i) + \dots + f_1(t_{i+L-1})f_2(t_{i+L-1}) = \langle f_{1,L}, f_{2,L} \rangle = 0, \forall i \in \overrightarrow{1, N-L+1}$$

С точки зрения динамических систем, между векторами задержек и многообразием траекторий скрытой системы существует диффеоморфизм, таким образом существование предполагаемого разделение говорит о разложимости скрытого многообразия в прямую сумму подпространств.

Как показано в подробной книге о методе SSA [10], предложенным способом невозможно строго разделить даже достаточно простые сигналы. Тем не менее, в асимптотическом приближении $N \rightarrow \infty, L \rightarrow \infty$ они разделяются, т.е. их корреляция в этом случае $\langle f_{1,L}, f_{2,L} \rangle \rightarrow 0$, см. табл. 1. Также там показано, что если $\exists \tilde{L} : \forall L > \tilde{L} \hookrightarrow \text{rank}(T) = r = \text{const}$, то исходный ряд в точности описывается авторегрессионной моделью.

Таблица 1: Асимптотическая разделимость компонент (Golyandina)

	const	cos	exp	exp (cos)	ak + b
const	-	+	+	+	-
cos	+	+	+	+	+
exp	+	+	+	+	+
exp (cos)	+	+	+	+	+
ak + b	-	+	+	+	-

Далее, любые данные содержат в себе ошибки (шум, неточности вычислений и т.д.), поэтому даже строго разделяемые ряды на практике не получится идеально декомпозировать. Поэтому применяется простая процедура: полученные в ходе разделения главных компонент X_1, X_2 *ганкелизуются*, т.е. каждая антидиагональ матриц усредняется и заменяется на это среднее. Таким образом оценивать полученную декомпозицию можно по получившейся невязке.

Что касается поиска группировки на основе SVD разложения, здесь не существует общего алгоритма. Часто для разделения главных компонент используют *относительную близость спектральных чисел*, группируя факторы по их близости друг к другу. В той же [10] есть обоснования такого подхода для нескольких классов функций.

Метод mSSA

Пусть теперь имеем m временных рядов $\{x_i(t_j)\}_{i=1}^m$, где $t_j \in \overrightarrow{1, N}$. Предполагая ту же гипотезу о природе порождения этих сигналов, будет действовать похожим на SSA способом. Для этого опять выбираем длину окна L и строим траекторные матрицы для

каждого ряда T_1, T_2, \dots, T_m . Теперь конкатенируем все матрицы в одну $T = [T_1 \ T_2 \ \dots \ T_m]$, которую далее раскладываем с помощью SVD:

$$T = \sum_i^r \sigma_i \mathbf{u}_i \mathbf{v}_i \Leftrightarrow \begin{cases} T_1 = \sum_i^r \sigma_i \mathbf{u}_i \mathbf{v}_i^1 \\ T_2 = \sum_i^r \sigma_i \mathbf{u}_i \mathbf{v}_i^2 \\ \dots \\ T_m = \sum_i^r \sigma_i \mathbf{u}_i \mathbf{v}_i^m \end{cases}$$

Здесь каждая главная компонента в пространстве строк \mathbf{v}_i представляется в виде конкатенации: $\mathbf{v}_i = (\mathbf{v}_i^1 \dots \mathbf{v}_i^m)$. Т.о. получаем некоторое разложение траекторных матриц, которое не является SVD-разложением в общем случае (т.к. компоненты \mathbf{v}_i^k не обязаны быть ортогональны между собой). Тем не менее базис столбцов у всех сигналов одинаковый $\{\mathbf{u}_i\}_{i=1}^r$ и ортонормированный, "сингулярные" числа тоже одинаковые. В итоге общее столбцовое пространство связывает траекторные матрицы каждого ряда и учитывает их общую природу порождения.

Дальнейший ход действий для декомпозиции аналогичен методу SSA. Можно рассматривать разложение каждой траекторной матрицы отдельно, но т.к. все они имеют общий набор $\{\sigma_i\}$, и группировка главных компонент часто делается на их основе, то разделение на группы происходит одинаковое для всех траекторных матриц. Получаем итоговое разложение каждого сигнала в виде:

$$\begin{aligned} x_1(t) &= \hat{f}_1(t) + \hat{f}_2(t) + \dots + \hat{f}_n(t) \\ x_2(t) &= \hat{g}_1(t) + \hat{g}_2(t) + \dots + \hat{g}_n(t) \\ &\dots \\ x_m(t) &= \hat{h}_1(t) + \hat{h}_2(t) + \dots + \hat{h}_n(t) \end{aligned}$$

Метод tSSA

Данный подход основан на сборке того же набора рядов $\{x_i(t_j)\}_{i=1}^m$ в тензор и применения уже *тензорного* разложения вместо рассматриваемого до этого SVD. Сразу отметим, что способов упаковки и разложений для тензоров можно предложить огромное количество (для базового ознакомления можно, например, обратиться к [9]; также см. предлагаемые работы в [Введение](#)), являясь ещё скудно изученной областью в приложении к временным рядам.

В данной работе предлагается вдохновлённый классическим SSA способ получения *траекторного тензора* \mathbf{T} : как и в mSSA, строятся траекторные матрицы для каждого ряда T_1, \dots, T_m , после чего данные матрицы состыковываются друг с другом по третьему измерению (индексу) тензора. Т.о. измерениям \mathbf{T} соответствуют столбцы и строки временных задержек, а также номер сигнала. Далее, к полученному тензору применяется *каноническое разложение* (Canonical polyadic decomposition, CPD), т.к. нам важно сохранить структуру суммы одноранговых факторов (см. рис.4).

С математической точки зрения это выглядит так:

$$\mathbf{T} = \sum_i^r \mathbf{a}_i \otimes \mathbf{b}_i \otimes \mathbf{c}_i \Leftrightarrow \begin{cases} T_1 = \sum_i^r \mathbf{c}_i[1] \cdot \mathbf{a}_i \mathbf{b}_i \\ T_2 = \sum_i^r \mathbf{c}_i[2] \cdot \mathbf{a}_i \mathbf{b}_i \\ \dots \\ T_m = \sum_i^r \mathbf{c}_i[m] \cdot \mathbf{a}_i \mathbf{b}_i \end{cases} \quad (1)$$

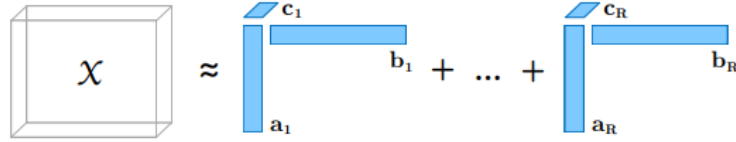


Рис. 4: Иллюстрация канонического разложения

Здесь записано разложение исходного тензора в сумму тензорных произведений набора векторов $\mathbf{a}_i, \mathbf{b}_i, \mathbf{c}_i$, которые можно для удобства состыковать в матрицы A, B, C . Если зафиксировать третий индекс, то получим разложение каждой траекторной матрицы T_i , где $\mathbf{c}_i[k]$ — это k -ая компонента вектора \mathbf{c}_i . Таким образом разложение происходит по одному базису как для столбцов, так и для строк у каждой матрицы задержек! Роль же сингулярных чисел играют строки матрицы C . Дальнейший ход действий для получения декомпозиции аналогичен SSA.

Полученное разложение обладает *рядом преимуществ* по сравнению с mSSA: теперь имеем общий базис как у столбцов, так и у строк матриц T_i , что ещё сильнее связывает каждый сигнал; тем более строки у T_i — это те же вектора задержек, только другой размерности, поэтому странно раскладывать их для каждой T_i по своему базису, если столбцы раскладываются по одному. Также данное разложение более гибкое, т.к. для каждой матрицы задержки имеется свой набор $\mathbf{c}_i[k], k \in \overrightarrow{1, r}$ "сингулярных" чисел, что позволяет каждому сигналу подстроиться под базисные векторы индивидуально. Таким образом каждую траекторную матрицу можно раскладывать также индивидуально, что было затруднительно в mSSA. Ещё одним положительным моментом является свойство CPD-разложения: при весьма необременительных условиях оно единственно, что сводит поиск любого разложения траекторных матриц в виде (1) к CPD-разложению тензора \mathbf{T} .

Тем не менее есть и некоторые недостатки по сравнению с матричным разложением: теперь базисы в пространстве столбцов не составляют ортогональную систему (чего не скажешь о mSSA), в пространстве строк аналогично (но у mSSA та же проблема). Также их необходимо нормировать для преемственности с предыдущими методами, нормы векторов скорректируют изначальные "сингулярные" числа (далее по умолчанию предполагаем нормированность). Ещё эти числа могут быть отрицательными, что уменьшает их интерпретируемость и помощь в группировке. Тем не менее теперь можно отбрасывать те из них, что будут *малы по модулю*.

Построение прогноза в рассмотренных методах

SSA

Опишем метод построения прогноза в модели **SSA**. Допустим последнее известное значение ряда x_T , хотим построить прогноз на k шагов вперёд.

Считаем, что ширина окна L для векторов задержки выбрана достаточная, а также имеем достаточное количество измерений ряда N , чтобы выполнялась теорема Такенса.

Тогда динамика векторов задержек диффеоморфна динамике скрытой порождающей наш ряд системе, т.е. все её траектории лежат в некотором многообразии $M \subset \mathcal{L}_r$, где $\mathcal{L}_r = \text{Lin}(\mathbf{u}_1, \dots, \mathbf{u}_r)$ - линейная оболочка ортонормированной системы, найденной с помощью SVD разложения. Введём матрицу $U = [\mathbf{u}_1, \dots, \mathbf{u}_r]$.

Теперь рассмотрим вектор задержки $\mathbf{x} = (x_{T+k-L+1} \dots x_T \dots x_{T+k})$; разделим его на неизвестную (последние k компонент) \mathbf{x}_{pr} и известную (оставшиеся компоненты) \mathbf{x}_{kn} часть. Аналогично разделим матрицу U на U_{kn} (первые $L - k$ строк) и U_{pr} (последние k строк). Имеем $\mathbf{x} \in M \subset \mathcal{L}_r$, т.е. этот вектор раскладывается по базису $\{\mathbf{u}_i\}^r$, а значит $\exists! \boldsymbol{\lambda} \in \mathbb{R}^r$ т.ч.:

$$\mathbf{x} = U\boldsymbol{\lambda} \Leftrightarrow \begin{cases} \mathbf{x}_{kn} = U_{kn}\boldsymbol{\lambda} \\ \mathbf{x}_{pr} = U_{pr}\boldsymbol{\lambda} \end{cases} \quad (2)$$

Это основное уравнение для получения предсказания, на его основе будет построен прогноз и в tSSA. Пока что заметим: из ортогональности U можно явно получить $\boldsymbol{\lambda} = U^T \mathbf{x}$. Подставим это в выражение для \mathbf{x}_{pr} :

$$\begin{aligned} \mathbf{x}_{pr} &= U_{pr} U^T \mathbf{x} = U_{pr} (U_{kn}^T U_{pr}^T) (\mathbf{x}_{kn} \mathbf{x}_{pr})^T = U_{pr} (U_{kn}^T \mathbf{x}_{kn} + U_{pr}^T \mathbf{x}_{pr}) \\ \mathbf{x}_{pr} (I - U_{pr} U_{pr}^T) &= U_{pr} U_{kn}^T \mathbf{x}_{kn} \\ \mathbf{x}_{pr} &= (I - U_{pr} U_{pr}^T)^{-1} U_{pr} U_{kn}^T \mathbf{x}_{kn} \end{aligned} \quad (3)$$

Последнее выражение верно при условии обратимости соответствующего члена.

Проанализируем ещё раз систему из (2): найти $\boldsymbol{\lambda}$ можно только из выражения для \mathbf{x}_{kn} ; сделав это, мы можем получить желаемый прогноз \mathbf{x}_{pr} из той же (2). Мы знаем, что $\text{rank}(U) = r$ и того же хочется для U_{kn} , ведь в противном случае ядро $\text{Ker}(U_{kn}) \neq \{0\}$ и даже если решение для $\boldsymbol{\lambda}$ есть, то оно не единственно и может быть сколь угодно большим: $\boldsymbol{\lambda} = \boldsymbol{\lambda}_{\text{particular}} + \mathbf{c} \cdot \text{Ker}(U_{kn})$, где \mathbf{c} — вектор произвольных констант. А в этом случае сколь угодно большим и неопределённым может быть и \mathbf{x}_{pr} , что плохо с точки зрения интерпретируемости (с другой стороны можно привлечь здесь идеи байесовского анализа и ввести априорное распределение на вектор \mathbf{c} , далее по имеющимся данным строить апостериорное распределение для \mathbf{x}_{pr} и делать прогноз, но это не тема данной работы). $\text{rank}(U_{kn})$ определяется тем, где находятся линейно-независимые строки в матрице U : хочется, чтобы все они находились как можно выше в U , чтобы U_{kn} была полного ранга даже для больших горизонтов прогнозирования k . Пусть в U нашли такой набор строк, причём наибольший номер строки в этом наборе получился K . Тогда понятно, что, во-первых, $K \geq r$, во-вторых, однозначный прогноз можно получить только для $k \leq L - K \leq L - r$.

Теперь обсудим существование решения для $\boldsymbol{\lambda}$. Если его нет, то нет и решения для \mathbf{x} , но мы предполагаем, что есть многообразие $M \subset \mathcal{L}_r$, в котором лежат траектории векторов задержек, и они должны быть определены для любых моментов времени. Вектор x содержит в себе часть вектора задержки и если решения для $\boldsymbol{\lambda}$ нет, то вообще не существует точки траектории, которую потенциально представляет \mathbf{x} (потенциально — потому что вообще он не определён полностью), и это противоречие. Таким образом, если все наши предположения, продиктованные теорией Такенса, выполнены, мы можем рассчитывать на *однозначно определённый прогноз* временного ряда. Отсюда же следует условие для обратимости матрицы $I - U_{pr} U_{pr}^T$.

Ещё одно замечание: из всего вышесказанного следует, что можно получать прогнозы разными способами, ведь k можно выбирать любым от 1 до $L - K$. Из однозначности этого прогноза для всех таких k следует, что все эти способы должны приводить к одному результату. Для вычисления \mathbf{x}_{pr} нужно обратить матрицу размера $k \times k$ и перемножить два

раза вектор на матрицу. Т.к. обращение в общем случае достаточно трудоёмкая операция, то рекомендуется строить прогноз для $k = 1$ столько раз, сколько нужно.

mSSA

Из всего вышесказанного, нетрудно понять, как строиться прогноз в **mSSA**: здесь мы также имеем ортонормированный базис в пространстве траекторий векторов задержек $\{\mathbf{u}_i\}^r$, единый для всех сигналов. Таким образом поступаем аналогичным SSA способом для каждого имеющегося временного ряда.

tSSA

Для **tSSA** ситуация несколько иная. Здесь роль U играет матрица A (см. раздел выше), которая полного ранга, но не ортогональная (но все столбцы нормированы в единицу по построению). Соответственно уравнение (2) здесь выглядит:

$$\mathbf{x} = A\boldsymbol{\lambda} \Leftrightarrow \begin{cases} \mathbf{x}_{kn} = A_{kn}\boldsymbol{\lambda} \\ \mathbf{x}_{pr} = A_{pr}\boldsymbol{\lambda} \end{cases} \quad (4)$$

Вышеизложенная теория полностью переносится на этот случай, за исключением того, что выразить $\boldsymbol{\lambda}$ через A просто не получится, но тем не менее решение должно существовать для любого горизонта прогнозирования $k \leq L - K$.

Разложим A через SVD: $A = U\Sigma V^T$, а также выразим $\boldsymbol{\lambda}$ через псевдообратную к A : $\boldsymbol{\lambda} = (A^T A)^{-1} A^T \mathbf{x}$ — в наших условиях это и есть решение СЛАУ выше (в общем случае — решение только в смысле наименьших квадратов). Проводя абсолютно аналогичные (3) операции, получим:

$$(I - A_{pr}(A^T A)^{-1} A^T A_{pr}^T) \mathbf{x}_{pr} = A_{pr}(A^T A)^{-1} A^T A_{kn}^T \mathbf{x}_{kn}$$

Также проведя простые операции с SVD разложением для A , получим следующее:

$$\begin{aligned} A_{pr}(A^T A)^{-1} A^T A_{pr}^T &= U_{pr}(U^T V \Sigma) U_{pr}^T \\ A_{pr}(A^T A)^{-1} A^T A_{kn}^T &= U_{pr}(U^T V \Sigma) U_{kn}^T \end{aligned}$$

Откуда получаем:

$$(I - U_{pr}(U^T V \Sigma) U_{pr}^T) \mathbf{x}_{pr} = U_{pr}(U^T V \Sigma) U_{kn}^T \mathbf{x}_{kn} \quad (5)$$

Получаем очень похожее на (3) выражение, в котором дополнительно фигурирует невырожденная квадратная матрица $U^T V \Sigma$ размера $r \times r$ (r — тензорный ранг). Домножая слева на обратную матрицу, можно выразить \mathbf{x}_{pr} , но, т.к. на практике достаточно и проще всего делать прогноз на $k = 1$, то можно из (4) выразить $\boldsymbol{\lambda} = (A_{kn}^T A_{kn})^{-1} A_{kn}^T \mathbf{x}_{kn}$, тогда получим:

$$\mathbf{x}_{pr} = A_{pr}(A_{kn}^T A_{kn})^{-1} A_{kn}^T \mathbf{x}_{kn} \quad (6)$$

Здесь $A_{pr}(A_{kn}^T A_{kn})^{-1} A_{kn}^T$ — вектор-строка, и, вычислив её один раз, можно быстро делать прогноз на много шагов вперёд.

Класс сигналов, описываемых методами типа SSA

Ещё раз обратим внимание на формулы прогноза (6) и (3) для горизонта прогнозирования $k = 1$. В этом случае слева стоит неизвестное значение ряда x_{T+n} , справа стоит умножение некоторого вектора строки $(h_1 \dots h_{L-1})$ на вектор-столбец известных значений ряда $(x_{T+n-L+1} \dots x_{T+n-1})^T$. Т.о. получаем:

$$x_{T+n} = \sum_{i=1}^{L-1} h_i \cdot x_{T+n-L+i} \quad (7)$$

Получаем, что предсказание методом SSA, по-сути, сводится к *авторегрессионной* модели ряда $AR(L-1)$. Соответственно, поведение прогноза полностью определяется коэффициентами h_i : ряд может неограниченно расти, затухать и/или иметь периодическую динамику. Таким образом найден класс сигналов, описываемых рассмотренными выше методами.

Практические трудности рассмотренных подходов

Проблема выбора группировки

Как и в обычном SSA, как и в других рассматриваемых методах, необходимо группировать полученную сумму факторов в разложении матриц T_i . Для SSA и mSSA это возможно делать по близости сингулярных чисел, хотя нет абсолютно никакой гарантии в правильности данного подхода. В tSSA из-за того, что эти числа могут быть отрицательными и более разреженными, такой метод отбора может быть более затруднительным. Качество полученной группировки можно оценить по совокупной ошибке при ганкелизации полученных факторов. Но предложить быстрый метод хорошего выбора на основании разложений пока затруднительно. Можно предложить полный перебор группировок: их потребуется рассмотреть порядка B_r , где B_j - числа Белла; а также метод локального поиска из предложенной начальной группировки: потребуется $(n-1)r$ просмотров в локальной окрестности, где m — количество групп в данной.

Сложность вычисления декомпозиций

Предложенные методы работают с разложением матриц и тензоров, что при больших объёмах выборок и выбора длины окна L может привести к невозможности их использования, по крайней мере стандартных алгоритмов. Например, в табл.5 приведено время работы и требования по памяти для SVD-разложения из библиотеки *scipy*. Поэтому для размеров временных рядов $\gtrsim 10^4$ необходимо искать быстрые приближённые решения.

Рис. 5: Сложность SVD-разложения (scipy)

n	10^3	10^4	10^5
Время	0.1 s	0.9 min	16 h
Mem(A)	8 Mb	0.8 Gb	80 Gb

Для CPD-разложения всё также непросто. Несмотря на его хорошие свойства, вычисление канонического ранга r является NP-трудной задачей. Поэтому при применении tSSA его необходимо подбирать. Из-за неточности этого подбора возможна большая ошибка аппроксимации исходного тензора \mathbf{T} , а выбор больших r приводит к огромному времени вычисления и потребляемой памяти. Достаточно быстрый приближённый алгоритм ALS

(Alternating Least Squares) всё равно оперирует матрицами, их свёртками и нахождением минимума матричных функций, что катастрофично для тензоров больших размеров.

Список литературы

- [1] Dimitrios Asteriou и Stephen Hall. «Vector Autoregressive (VAR) Models and Causality Tests». В: ЯНВ. 2016, с. 333—346. ISBN: 978-1-137-41546-2. DOI: [10.1057/978-1-137-41547-9_15](https://doi.org/10.1057/978-1-137-41547-9_15).
- [2] Robert B. Cleveland и др. «STL: A Seasonal-Trend Decomposition Procedure Based on Loess (with Discussion)». В: *Journal of Official Statistics* 6 (1990), с. 3—73.
- [3] Estela Dagum и Silvia Bianconcini. *Seasonal Adjustment Methods and Real Time Trend-Cycle Estimation*. АВГ. 2016. ISBN: 978-3-319-31820-2.
- [4] W. Enders и John Wiley & Sons. *Applied Econometric Time Series*. Wiley series in probability and statistics. Wiley, 2010. ISBN: 9788126543915. URL: <https://books.google.ru/books?id=vzJ0CgAAQBAJ>.
- [5] C. W. J. Granger. «Investigating Causal Relations by Econometric Models and Cross-spectral Methods». В: *Econometrica* 37.3 (1969), с. 424—438. ISSN: 00129682, 14680262. URL: <http://www.jstor.org/stable/1912791> (дата обр. 24.01.2024).
- [6] Hansika Hewamalage, Christoph Bergmeir и Kasun Bandara. «Recurrent Neural Networks for Time Series Forecasting: Current status and future directions». В: *International Journal of Forecasting* 37.1 (2021), с. 388—427. ISSN: 0169-2070. DOI: <https://doi.org/10.1016/j.ijforecast.2020.06.008>. URL: <https://www.sciencedirect.com/science/article/pii/S0169207020300996>.
- [7] Sepp Hochreiter и Jürgen Schmidhuber. «Long Short-Term Memory». В: *Neural Computation* 9.8 (нояб. 1997), с. 1735—1780. ISSN: 0899-7667. DOI: [10.1162/neco.1997.9.8.1735](https://doi.org/10.1162/neco.1997.9.8.1735). eprint: <https://direct.mit.edu/neco/article-pdf/9/8/1735/813796/neco.1997.9.8.1735.pdf>. URL: <https://doi.org/10.1162/neco.1997.9.8.1735>.
- [8] Robin John Hyndman и George Athanasopoulos. *Forecasting: Principles and Practice*. English. 2nd. Australia: OTexts, 2018.
- [9] Stephan Rabanser, Oleksandr Shchur и Stephan Günnemann. *Introduction to Tensor Decompositions and their Applications in Machine Learning*. 2017. arXiv: [1711.10781](https://arxiv.org/abs/1711.10781) [stat.ML].
- [10] D. Stepanov и N. Golyandina. «SSA-based approaches to analysis and forecast of multidimensional time series». не определен. В: *Proceedings of the 5th St.Petersburg Workshop on Simulation*. 2005, с. 293—298.
- [11] George Sugihara и др. «Detecting Causality in Complex Ecosystems». В: *Science* 338.6106 (окт. 2012), с. 496. DOI: [10.1126/science.1227079](https://doi.org/10.1126/science.1227079).
- [12] Floris Takens. «Detecting Strange Attractors in Turbulence». В: *Dynamical Systems and Turbulence, Warwick 1980*. Под ред. David Rand и Lai-Sang Young. Т. 898. Lecture Notes in Mathematics. Berlin: Springer, 1981. Гл. 21, с. 366—381. ISBN: 978-3-540-11171-9. DOI: [10.1007/bfb0091924](https://doi.org/10.1007/bfb0091924). URL: <http://dx.doi.org/10.1007/bfb0091924>.
- [13] Ahmed Tealab. «Time series forecasting using artificial neural networks methodologies: A systematic review». В: *Future Computing and Informatics Journal* 3.2 (2018), с. 334—340. ISSN: 2314-7288. DOI: <https://doi.org/10.1016/j.fcij.2018.10.003>. URL: <https://www.sciencedirect.com/science/article/pii/S2314728817300715>.
- [14] Granville Tunncliffe Wilson. «Time Series Analysis: Forecasting and Control, 5th Edition, by George E. P. Box, Gwilym M. Jenkins, Gregory C. Reinsel and Greta M. Ljung, 2015. Published by John Wiley and Sons Inc., Hoboken, New Jersey, pp. 712. ISBN: 978-1-118-67502-1». В: *Journal of Time Series Analysis* 37 (март 2016), n/a—n/a. DOI: [10.1111/jtsa.12194](https://doi.org/10.1111/jtsa.12194).

- [15] Arnold Zellner. «An Efficient Method of Estimating Seemingly Unrelated Regressions and Tests for Aggregation Bias». В: *Journal of the American Statistical Association* 57.298 (1962), с. 348—368. DOI: [10.1080/01621459.1962.10480664](https://doi.org/10.1080/01621459.1962.10480664). eprint: <https://www.tandfonline.com/doi/pdf/10.1080/01621459.1962.10480664>. URL: <https://www.tandfonline.com/doi/abs/10.1080/01621459.1962.10480664>.
- [16] Xueli Zhang и др. «Robust recurrent neural networks for time series forecasting». В: *Neurocomputing* 526 (2023), с. 143—157. ISSN: 0925-2312. DOI: <https://doi.org/10.1016/j.neucom.2023.01.037>. URL: <https://www.sciencedirect.com/science/article/pii/S0925231223000462>.
- [17] В. Вольтерра. «Matematicheskaya teoriya bor'by za sushchestvovanie». В: *Usp. Fiz. Nauk* 8.1 (1928), с. 13—34. DOI: [10.3367/UFNr.0008.192801b.0013](https://doi.org/10.3367/UFNr.0008.192801b.0013). URL: <https://ufn.ru/ru/articles/1928/1/b/>.