

# Сингулярный спектральный анализ: введение

Чуйкин Никита

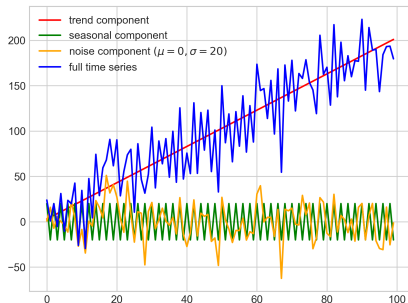
Научный руководитель: Егорова Людмила Геннадьевна

March 2024

- ① Основы SSA
- ② Области применения
- ③ Примеры на данных
- ④ Для дальнейшего изучения
- ⑤ Источники

# Алгоритм SSA: введение

Пусть дан одномерный временной ряд  $(f_1, f_2 \dots f_n)$ ,  $f_i \in \mathbb{R}$ . Наша цель - разложить его на компоненты  $f(x) = g(x) + p(x) + \epsilon$ , где  $g(x)$  - тренд,  $p(x)$  - сезонность,  $\epsilon$  - случайная компонента.



Попробуем получить информацию о корреляции частей временного ряда друг с другом.

# Алгоритм SSA: агрегирование

Выберем длину окна (*window length*)  $l$ , будем двигать этот вектор по временному ряду и составим матрицу

$$S = \begin{pmatrix} f_1 & f_2 & f_3 & \dots & f_{n-l} \\ f_2 & f_3 & f_4 & \dots & f_{n-l+1} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ f_l & f_{l+1} & f_{l+2} & \dots & f_n \end{pmatrix}$$

## Определение

Так построенную матрицу будем называть траекторной (*trajectory*).

$$S \in \mathbb{R}^{l \times (n-l+1)}$$

Все траекторные матрицы имеют **ганкелеву форму**, т.е. содержат одинаковые элементы на побочных диагоналях.

# Иллюстративный пример

Пусть зависимость описывается простой линейной формулой  $y = 2x$ ,  $x \in 0, \dots, 4$ . Тогда временной ряд  $y = (0, 2, 4, 6, 8)$ .

Составим траекторную матрицу для длины окна  $l = 2$ :

$$S = \begin{pmatrix} 0 & 2 & 4 & 6 \\ 2 & 4 & 6 & 8 \end{pmatrix}$$

## Определение

SVD (singular value decomposition) матрицы  $A \in \mathbb{R}^{n \times m}$  - это представление матрицы в форме  $A = U\Sigma V$ , где  $U \in \mathbb{R}^{n \times n}$  и  $V \in \mathbb{R}^{m \times m}$  - ортогональные матрицы, а  $\Sigma \in \mathbb{R}^{n \times m}$  - диагональная матрица.

На диагонали матрицы  $\Sigma$  в невозрастающем порядке стоят сингулярные числа  $\sigma_1, \sigma_2 \dots \sigma_r$ , где  $r$  - ранг матрицы  $A$ . Впредь мы будем использовать только низкоранговое SVD для заданного  $r$ , i.e. "отрезать" последние столбцы матрицы  $U$  и строки матрицы  $V$ .

# SVD факторизация: как посчитать вручную

Заметим, что  $A^T A = V^T \Sigma^T U^T U \Sigma V = V^T \Sigma^2 V$ . То есть, матрица  $V$  по строкам содержит собственные векторы  $A^T A$ , соответствующие квадрату сингулярных чисел.

## Шаг 1: матрица $\Sigma$

Составим матрицу  $A^T A$ . Найдём её собственные значения  $\lambda_1, \lambda_2 \dots \lambda_r$ .  
Сингулярные числа  $\sigma_1 = \sqrt{\lambda_1}, \dots, \sigma_r = \sqrt{\lambda_r}$

## Шаг 2: матрица $V$

Находятся собственные векторы матрицы  $A^T A$  и укладываются в строки матрицы  $V$

## Шаг 3: матрица $U$

Решаем систему линейных уравнений  $A = U \Sigma V$ . Или в более удобном виде  $A^T U = V^T \Sigma$

# SVD факторизация: свойства

## Определение

Пусть  $A \in \mathbb{R}^{n \times n}$ . Нормой Фробениуса будем называть

$$\|A\|_F = \sqrt{\sum_{i,j} a_{ij}^2}$$

## Теорема Эккарта-Янга

Матрица, построенная через малоранговое SVD является ближайшей среди матриц ранга  $r$  к матрице  $A$  по норме Фробениуса.

Разумно ожидать, что SVD траекторной матрицы будет выделять важные компоненты ряда. Используем эту идею.



# Алгоритм SSA: SVD приближение

Разложим траекторную матрицу  $S$  с помощью малорангового SVD:  
 $S = U\Sigma V = \sum_{i=1}^r \sigma_i u_i v_i^T$ ,  $u_i, v_i^T$  - векторы столбцы.

Теперь сгруппируем эти тройки (*eigentriplets*) в массивы  $l_1, l_2 \dots l_k$  и обозначим  $X_j = \sum_{i \in l_j} \sigma_i u_i v_i^T$ . Каждая такая траекторная матрица представляет собой компоненту временного ряда (?).

Как получить исходный ряд из матрицы  $S$ ? Достаточно найти среднее по всем побочным диагоналям матрицы (*hankelization*).

**Свойство оптимальности:** траекторная матрица для ряда, полученного из  $Y$  ганкелизацией является ближайшей с точки зрения нормы Фробениуса к  $Y$  из всех ганкелевых (Н.Голяндина, В.Некруткин, Д.Степанов 2003)

## Иллюстративный пример: продолжение

$$S = \begin{pmatrix} 0 & 2 & 4 & 6 \\ 2 & 4 & 6 & 8 \end{pmatrix}$$

SVD полного ранга будет иметь вид

$$S \approx \begin{pmatrix} -0.58 & 0.82 \\ 0.82 & -0.58 \end{pmatrix} \begin{pmatrix} 12.8 & 0 \\ 0 & 1.42 \end{pmatrix} \begin{pmatrix} -0.13 & 0.35 & -0.5 & -0.78 \\ -0.82 & -0.49 & 0.26 & 0.18 \end{pmatrix}$$

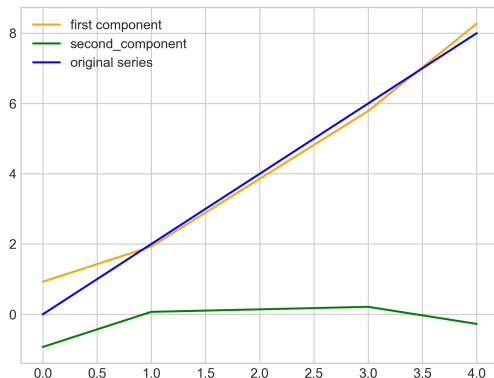
$$u_1 = \begin{pmatrix} -0.58 \\ 0.82 \end{pmatrix}, v_1 = (-0.13 \quad 0.35 \quad -0.5 \quad -0.78)$$

$$u_2 = \begin{pmatrix} 0.82 \\ -0.58 \end{pmatrix}, v_2 = (-0.82 \quad -0.49 \quad 0.26 \quad 0.18)$$

# Иллюстративный пример: сборка

$$y_1 = \text{Hankelization}(\sigma_1 u_1 v_1) = (0.94, 1.94, 3.65, 5.5, 8.15)$$

$$y_2 = \text{Hankelization}(\sigma_2 u_2 v_2) = (-0.94, -0.06, 0.35, 0.5, -0.15)$$



# Алгоритм SSA: резюме

- 1 Составляем траекторную матрицу исходного временного ряда  $(f_1, f_2, \dots, f_n)$
- 2 Получаем SVD разложение и группируем полученные компоненты сингулярного разложения
- 3 В каждой группе складываем траекторные матрицы и проводим ганкелизацию для получения временного ряда

# Скорость вычислений

Посчитаем наихудшее время исполнения ( $I = \frac{n}{2}$ ) с помощью прямых вычислений. Подробнее - (*Anton Korobeynikov, 2010*).

- **Агрегирование:** незначительно
- **SVD:**  $O(n^3)$
- **Группировка:**  $O(n^3)$
- **Генкелизация:**  $O(n^3)$  сложений и  $O(n^2)$  умножений

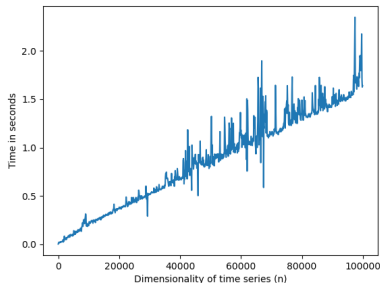
Но! Генкелевы матрицы можно привести к Тёплицевым и объединить этап группировки с генкелизацией. За счёт того, что на Тёплицеву матрицу можно умножать за  $O(n \log n)$ , два этих этапа будут считаться в той же ассимптотике. Если мы дополнительно знаем, сколько собственных троек необходимо ( $r$ ), то **общее время вычислений (из-за усеченного SVD) можно сократить до  $O(rn \log n)$ .**

# Скорость вычислений: пример

## Утверждение

Скорость вычисления SVD разложения =  $O(ln^2)$ , если  $l \geq (n - l + 1)$ , и  $O(nl^2)$  - иначе. (Trefethen N., Bau D. *Numeric linear algebra*)

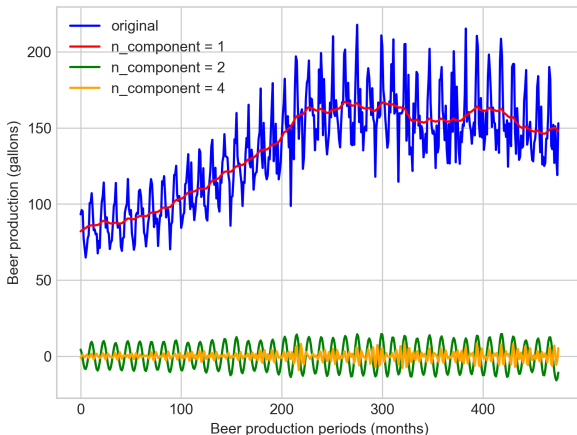
Временной ряд длины  $n$  генерировался из нормального распределения с  $l = 4$ . Скорость всего алгоритма SSA так же оказалась линейна для случая  $l \leq (n - l + 1)$ .



- Очистка данных от шума
- Интерполяция
- Предсказания на основе линейной рекуррентной формулы
$$f_k = \sum_{i=n-l}^{k-1} a_i f_i$$
- Выделение отдельных компонент ряда

# Пример на данных: как ведут себя компоненты

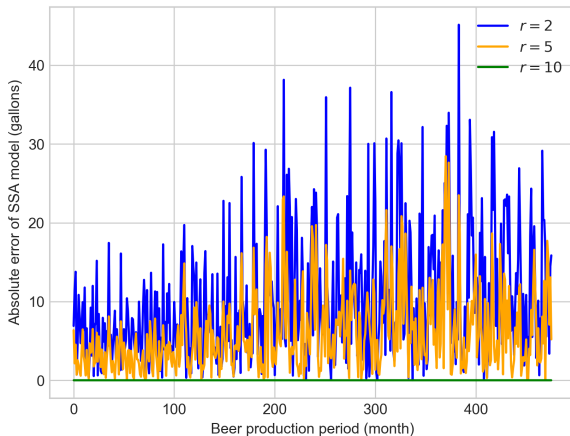
Данные о производстве пива в США: [kaggle.com](https://www.kaggle.com/datasets/kaggle/beer-production). График показывает, как ведут себя различные компоненты соответствующие 1,2 и 4 собственным тройкам для SSA с параметром  $l = 10$ .





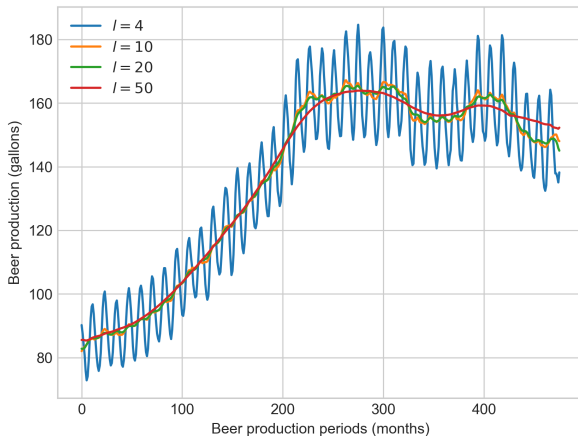
# Пример на данных: как ведет себя ошибка

Теперь посмотрим на ошибку аппроксимации ( $r$  - ранг разложения).



# Пример на данных: выбор длины окна

Посмотрим на то, как изменяется первая компонента разложения при изменении длины окна.



# Для дальнейшего изучения: группировка

## Определение

Временные ряды  $x_1, x_2 \in \mathbb{R}^n$ , полученные из траекторных матриц  $X_1, X_2 \in \mathbb{R}^{l \times n-l+1}$  слабо  $L$ -разделимы, если пространства столбцов (*column spaces*) и строк (*row spaces*) матриц  $X_1$  и  $X_2$  ортогональны.

## Утверждение

Пусть  $x_1, x_2 \in \mathbb{R}^n$ ,  $x = x_1 + x_2$  и  $x_1, x_2$  слабо  $L$ -разделимы. Тогда для ряда  $x$  существует такое сингулярное разложение траекторной матрицы  $X$ , что его можно разбить на две части, являющиеся траекторными матрицами  $x_1, x_2$

Но  $L$ -разделимость слишком сильное требование, которое нуждается в ослаблениях. Другие возможные варианты: автокорреляция, близость сингулярных значений, прокси-показатель ортогональности (например, скалярное произведение), асимптотическая разделимость.

# Для дальнейшего изучения: многомерный SSA (MSSA)

Пусть есть два временных ряда  $f_1, f_2$  - как учесть не только автокорреляцию, но и корреляцию между рядами? Есть несколько способов для анализа двумерных рядов:

- Раскладывать ряды по отдельности.
- Составить траекторную матрицу как  $S = [S_1 : S_2]$ ,  $S_{1,2}$  — траекторная матрица соответствующего ряда.
- Комплексное SSA. Конструируется новый временной ряд  $c = (f_1 + ig_1, \dots, f_n + ig_n)$ . Для него аналогично считается SVD.

Все три способа значительно отличаются по распределению компонент рядов.

## Для дальнейшего изучения: зависимость от будущего

Пусть мы имеем весь временной ряд  $(f_1, \dots, f_n)$ . Тогда насколько будут отличаться компоненты разложения для ряда  $(f_1, \dots, f_k)$ , где  $k < n$  - заданная точка отсечки, и полного ряда?

Оказывается, что по крайней мере первая компонента меняется лишь на первых и последних наблюдениях из временного ряда. Gif демонстрирует зависимость первой компоненты от  $k$ . По оси ОХ - время до отсечки  $k$ , по оси ОУ - значения временного ряда.

Эта задача аналогична задаче изучения эффективного SSA для большого ряда (когда  $n$  настолько велико, что вычислительных мощностей не хватает для вычисления SVD).

- Сайт, посвященный SSA
- Broomhead D.S., Ging G.P. *Extracting qualitative dynamics from experimental data*, 1986
- Golyandina N., Nekrutkin V., Zhigljavsky A. *Analysis of Time Series Structure*, 2001
- Голяндина Н., Некруткин В., Степанов Д. *Варианты метода «Гусеница»-SSA для анализа многомерных временных рядов*, 2003
- Вохмянин С. *Метод "Гусеница-SSA" как инструмент прогнозирования состояния финансового рынка*, 2010
- Голяндина Н. *Метод "Гусеница SSA": анализ временных рядов*, 2004
- Korobeynikov A. *Computation- and Space-Efficient Implementation of SSA*, 2010