Выявление малых слагаемых временного ряда

Г.В. Орлов

Федеральное государственное унитарное предприятие «Российский Федеральный Ядерный Центр-Всероссийский НИИ технической физики имени академика Е.И. Забабахина», Снежинск, Россия

Отправной точкой нынешнего доклада стала задача выявления резонансов штреков финского подземного рудника Полвиярви из данных работы 45 трехкомпонентных станций европейского проекта COGITO-MIN, расположенных на участке размерами $\approx 1.5 км \times \approx 1 км$, которые в течение месяца осуществляли запись сейсмических сигналов с частотой 500 опросов в секунду [1], беспрерывно выдавая в течении суток $43\,200\,000$ чисел только одной компоненты из трех. Все это время рудник круглосуточно работал, производя взрывные работы и добывая руду на глубине от 190 до 800 метров в разветвленной системе штреков, что создавало значительный сейсмический фон, в котором аддитивные сейсмические сигналы от резонансов штреков рудника составляют очень, очень малый уровень.

Исходной точкой решения задачи является алгоритм надежного выявления малых регулярных гармонических сигналов из суммы значительного числа регулярных и случайных составляющих, создаваемых механизмами работающего горного предприятия. Изучение десятков алгоритмов, существующих в бурно развивающейся науке обработки данных DataScience, показало, что наиболее перспективным методом выявления малых аддитивных регулярных составляющих и отделения их от шума является метод сингулярного разложения SVD (Singular Value Decomposition), который раскладывает двумерную матрицу X в сумму

$$X = \sum_{i=1}^{I} X_i, \quad I = rang(X) \qquad (1)$$

матриц X_i ранга 1, каждой из которых соответствует левый сингулярный вектор $\vec{u}_i$, норма которого s_i отражает вклад матрицы X_i в общую сумму.

Чтобы воспользоваться алгоритмом SVD для разложения **одномерных** сигналов коллективом математиков Gistat Group, возглавляемым А.А. Жиглявским и Н.Э. Голяндиной [1] из Санкт-Петербургского государственного университета, была реализована идея А.Н. Колмогорова создать из одномерного временного ряда $u = (u_0, u_1, \ldots, u_{N-1})$ $N > 2$ ганкелеву двумерную матрицу H с постоянными значениями вдоль побочных диагоналей, каждая колонка которой определяется значениями ряда u_n в скользящем вдоль этого ряда временном окне заданной длины L, сдвигаемом на одно значение от предыдущей колонки:

$$H=H(u,L,K)=\begin{pmatrix} u_0 & u_1 & u_2 & \cdots & u_{K-2} & u_{K-1} \\ u_1 & u_2 & u_3 & \cdots & u_{K-1} & u_K \\ u_2 & u_3 & u_4 & \cdots & u_K & u_{K+1} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ u_{L-1} & u_L & u_{L+1} & \cdots & u_{N-2} & u_{N-1} \end{pmatrix} \quad (2)$$

Далее, разлагая полученную матрицу H методом SVD в сумму элементарных матриц X_j

$$H = X_1+...+X_L, \quad X_j=\vec{x}_i\otimes\vec{y}_i^T, \ i=1,...,L, \quad (3)$$

каждая из которых определяется внешним произведением левого $\vec{x}_i$ и правого $\vec{y}_i$ собственных ее векторов, и аппроксимируя (ганкелизируя) их побочные диагонали значением в некоторой метрике (L_2, L, C), можно получить приближенное представление исходного ряда u в виде суммы L элементарных компонент длины N , вклад каждой из которых в общую сумму определяется величиной сингулярного значения s_i матрицы X_i. В России этот метод получил название метода главных компонент, а в Великобритании и США на него ссылаются под именем SSA (Singular Spectrum Analysis). Для большинства интересных на практике случаев такое разложение дает хорошую адекватность и высокую точность, что обеспечивают многочисленные теоремы из [1].

На рубеже веков по проекту МНТЦ мной был успешно применен метод SSA с программами от GistatGroup с целью нахождения электромагнитного импульса от мощного коротко замедленного взрыва на открытом руднике Курской магнитной аномалии в условиях электромагнитного шума, уровень которого превышал величину искомого сигнала **в сотни раз**. Был выявлен четкий сигнал ожидаемой формы и величины, с априори известной задержкой по времени.

В модуле numpy современного языка Python имеется высоко эффективная векторная программа numpy.linalg.svd, осуществляющая разложение двумерной матрицы X в сумму $X=X_1+X_2+...+X_I$ элементарных, взаимно ортогональных матриц X_i ранга 1 и выдающая матрицу U из столбцов левых собственных векторов матриц X_i , матрицу V из строк их правых собственных векторов, и вектор S собственных (сингулярных) чисел этих матриц. Эффективность этой программы поражает - на процессоре с четырьмя ядрами она раскладывает матрицу размерности 150000 на 20000 за несколько минут.

А вот ганкелизация по побочным диагоналям 20000 матриц X_i размерности 20000×150000 занимает неподъемное время — несколько суток. Изучение этой проблемы показало, что секретом быстрой работы программы svd является использование современного

западного индустриального стандарта создания программ LLVM, который в нашем институте практически неизвестен и который позволяет осуществить автоматическое высокоэффективное распараллеливанием на любой комбинации процессоров и их ядер. На языке Pyhon вся работа по ускорению счета свелась к написанию двух строк-

```
numba.config.NUMBA_NUM_THREADS=4
@numba.njit(parallel=True)
```
(4)

что ускорило счет более чем в 100 раз.

Еще одной проблемой в методе SSA, требующей значительных усилий, является проблема отбора нужных аддитивных компонент из полученных после разложения тысяч элементарных компонент. На языке Python был написан ряд программ, предлагающих эффективный и удобный сервис отбора таких элементарных компонент, которые адекватно соответствуют как временной, так и спектральной априорной о них информации.

Приведем наглядные тестовые примеры решения задач выявления слагаемых временного ряда методикой SSA. На отрезке времени [0,1] зададим массив $f(t)$ из N равномерно расположенных точек, для создания которого определим три функции:

$$\begin{gathered} t=(0,1,2,\ldots,N-1)/N\ , \\ f_1(t)=A_1\sin(2\pi a_1 t+\varphi_1),\ \ A_1=2,\ a_1=5,\ \ \varphi_1=\pi/7\ , \\ f_2(t)=A_2\sin(2\pi a_2 t+\varphi_2),\quad A_2=1,\ a_2=20,\varphi_2=\pi/3\ , \\ f_3(t)=a_3\exp(b_3 t),\ \ a_3=0.2,\ b_3=4\ . \end{gathered} \tag{5}$$

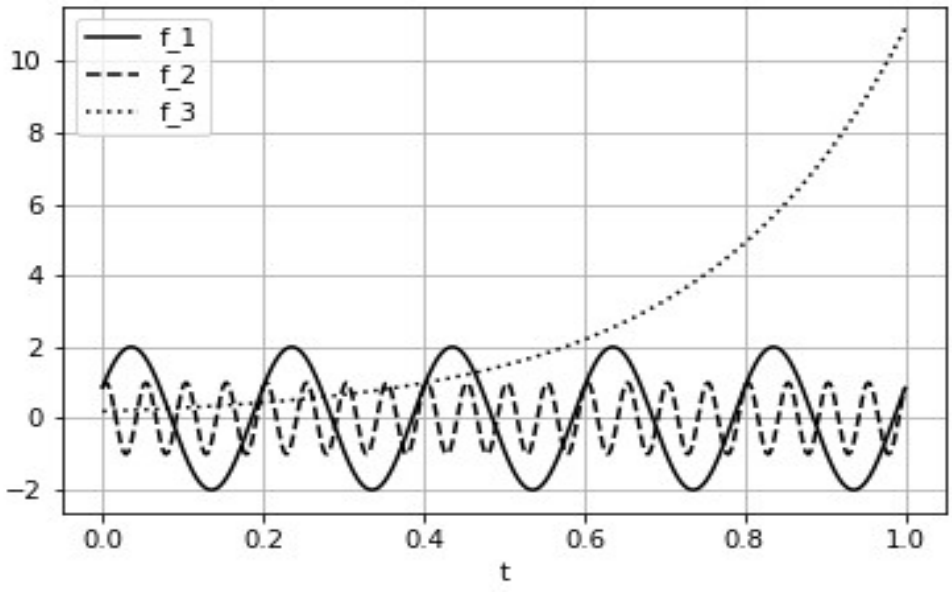

Рисунок 1 — Графики функций $f_1(t),f_2(t),f_3(t)$.

Зададим $N=100$, а параметр окна $L=N//2=50$, где через // обозначена как в языке Python операция деления нацело. Заметим, что максимальное значение параметра L метода SSA есть $L=N//2$. Выявим в ряде $f(t)=f_1(t)+f_2(t)$ его слагаемые $f_1(t), f_2(t)$. Сначала получим из ряда f ганкелеву матрицу H, главная диагональ которой есть ряд f , и разложим ее методом SVD выполним строку

```
U,S,V,timeSVD = HankSVD(f_1+f_2,L=N//2,printTime=True)
```
(6)

Очевидно, что элементарные матрицы X_l не являются ганкелевыми, хотя и сохраняют в себе неким образом это свойство, поскольку их сумма - матрица H - была ганкелевой. Для каждой матрицы разложения определим ее одномерный элементарный ряд h_l , значения которого есть среднее значение соответствующей побочной диагонали в норме L_2 . Занесем L таких рядов в матрицу h -

```
h = seriesL2_(U,S,V)
```
(7)

Выдадим картинки первых 24-х элементарных компонент с указанием индекса компоненты и ее процентного вклада :

```
pcts(h,S,N=24)
```
(8)

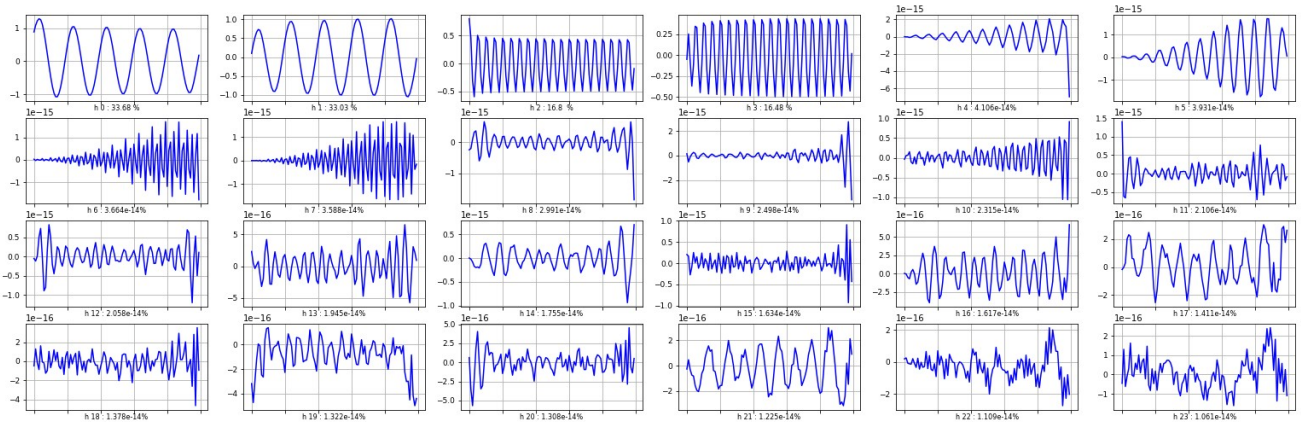

Рисунок 2 — Лист первых 24 элементарных компонент функции $f(t)=f_1(t)+f_2(t)$

Легко заметить, что первые 4 компоненты полностью определяют ряд, так как процентный вклад остальных компонент составляет величину ~10^{-14}. Также легко видеть, что первые две компоненты есть аддитивные составляющие функции f_1 с числом колебаний 5, а следующие две компоненты — аддитивные составляющие функции f_2 с числом колебаний 20. Видим, что метод SSA хорошо выявляет гармонические слагаемые, описывая каждое такое слагаемое двумя рядом стоящими элементарными компонентами.

Аналогично для функции $f(t)=f_1(t)+f_2(t)+f_3(t)$:

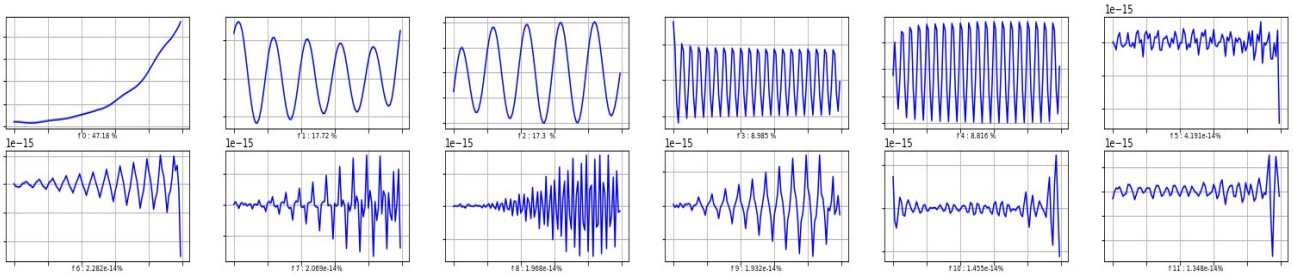

Рисунок 3 — Лист первых 12 элементарных компонент функции $f(t)=f_1(t)+f_2(t)+f_3(t)$

Для задачи фильтрации шума добавим к тестовой сумме $f(t)=f_1(t)+f_2(t)+f_3(t)$ постоянный гауссовский шум с нулевым средним и дисперсией $\sigma=10$, обозначив полученный временной ряд через `fn`, и выдадим обзорный график функций f(t) и fn(t):

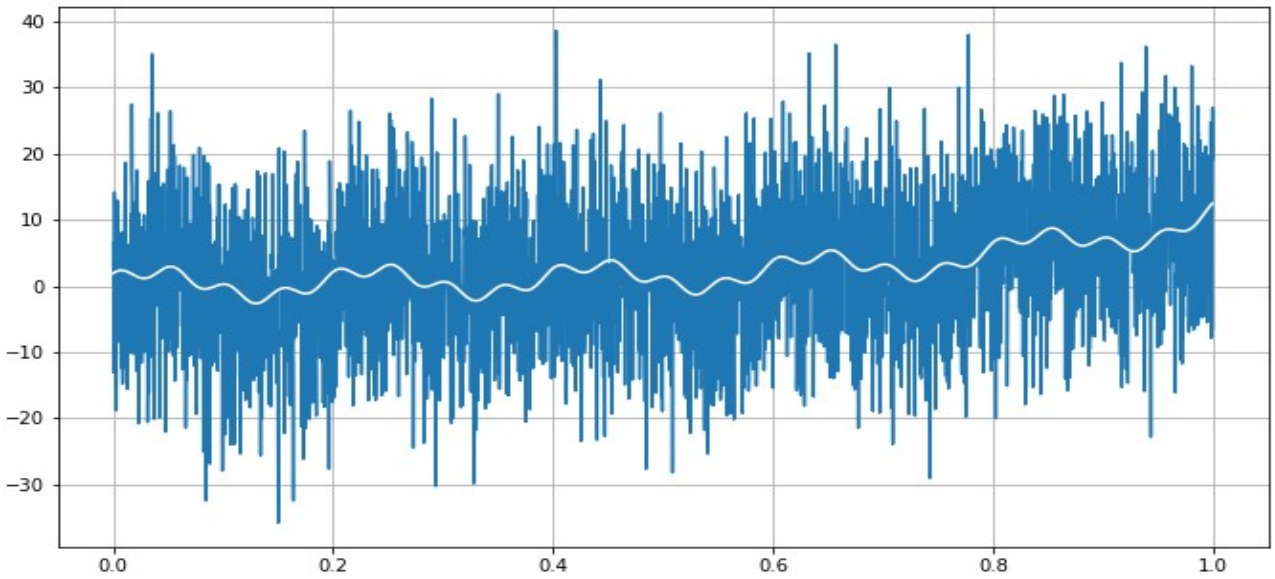

Рисунок 1.4 — Сравнительные графики функций fn(t) и f(t) (белая кривая)

Применим процедуру SSA к функции `fn` для N=3000, L=N//2=1500, вычислив и выдав 24 картинки его элементарных аддитивных составляющих:

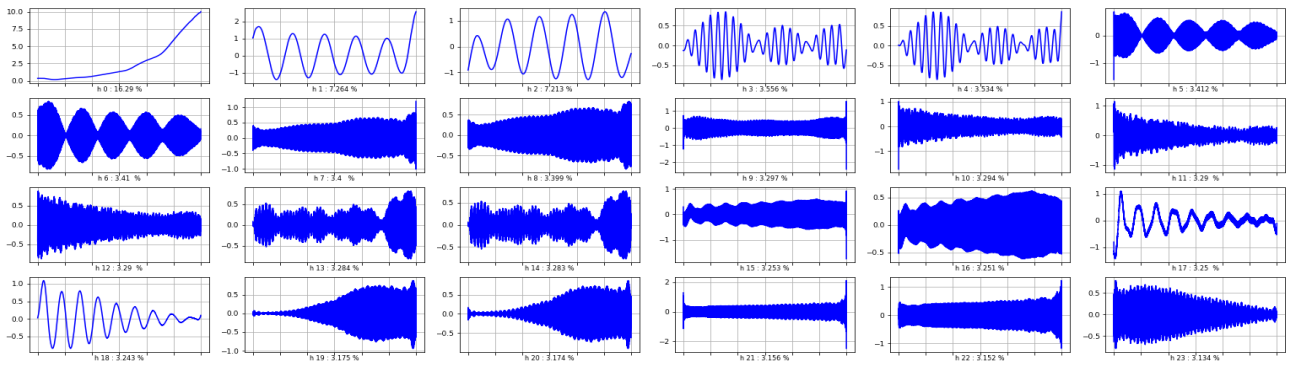

Рисунок 5 — Лист первых 24-х элементарных компонент функции `fn`

Замечаем, что первые пять компонент имеют регулярный характер и соответствуют исходным компонентам f_1, f_2, f_3 . Компонента h[18] также имеет регулярный характер, но явно видно, что ее гармоническая пара h[17] имеет нерегулярную аддитивную составляющую. С целью фильтрации шума у компоненты h[17] применим к ней метод SSA и выведем на совместный график исходную функцию $f(t)=f_1(t)+f_2(t)+f_3(t)$, ее приближение в виде суммы ее первых пяти элементарных компонент, а также эту аппроксимацию с добавкой еще двух регулярных элементарных компонент - очищенной компоненты 17 и старой компоненты 18 :

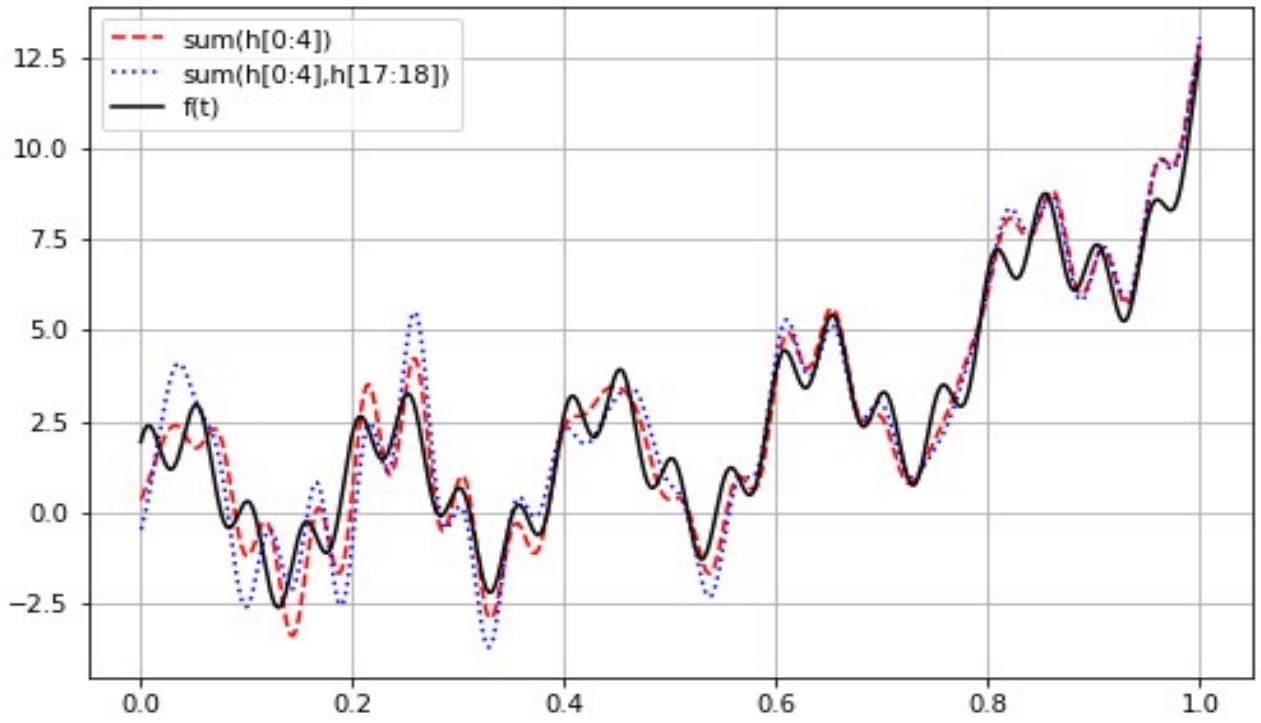

Рисунок 6 - Исходный ряд f(t)=f_1(t)+f_2(t)+f_3(t) и два его элементарных приближения

В задаче о резонансах штреков рудника Полвиярви была разработана модель резонанса границы цилиндрической полости. Эта модель указала резонансные частоты штреков. Для огромного подземного зала удалось выявить резонансный отклик в виде суммы трех резонансных частот. Понятно, что отбор резонансных компонент происходил автоматически по преобразованию Фурье компонент. Некоторые резонансные компоненты имели порядковый индекс 5079 с вкладом 0.1 % .

Список литературы

1. Голяндина Н.Э., Некруткин В.В., Браулов К.А. Метод „Гусеница“-SSA: анализ временных рядов. - Gistat Group, http://www.gistatgroup.com/gus/. - 2002.