Снижение размерности в задачах декодирования временных рядов и прогнозирования*

Р. В. Исаченко[1], М. Р. Владимирова[2], В. В. Стрижов[3]

Аннотация: Решается задача обнаружения зависимостей в прогнозируемой переменной. Вместо построения прогноза одного момента времени предлагается прогнозировать многомерный вектор прогноза. Рассматривается линейная модель метода частных наименьших квадратов. Описан метод нахождения матрицы совместного описания для исходных матриц объектов и ответов. Найденное описание является низкоразмерным, что позволяет построить устойчивую, простую модель. Проводится вычислительный эксперимент на реальных данных объемов потребления электроэнергии и данных сигналов кортикограмм.

Ключевые слова: декодирование временных рядов; метод частных наименьших квадратов; PLS; снижение размерности

1 Введение

В работе рассматривается задача восстановления зависимости между независимой и прогнозируемой переменными. Предлагаемая модель позволяет восстанавливать зависимость в случае многомерной прогнозируемой переменной. В случае задачи прогнозирования пространства объектов и ответов имеют одну и ту же природу. Для построения модели по исходному временному ряду строятся авторегрессионные матрицы объектов и ответов. Объектом является локальная история сигнала, ответом — значение сигнала в следующие моменты времени. Авторегрессионная модель предполагает, что значение сигнала в данный момент времени линейно зависит от предыдущих значений этого же сигнала.

В задаче декодирования временных рядов пространство объектов имеет значимо большую размерность. В этом случае пространства объектов и ответом имеют разную

*Работа выполнена при финансовой поддержке РФФИ (проект 16-07-01155).

[1]Московский физико-технический институт, isa-ro@yandex.ru

[2]Московский физико-технический институт, vladimirova.maria@phystech.edu

[3]Вычислительный центр им. А. А. Дородницына ФИЦ ИУ РАН, strijov@ccas.ru

природу, ответом является отклик системы на входной сигнал. Авторегрессионная матрица объектов содержит локальную историю входного сигнала, а авторегрессионная матрица ответов содержит локальную историю отклика.

Исходное признаковое пространство является многомерным. Избыточная размерность признакового пространства приводит к неустойчивости модели. Для решения этой проблемы используются методы отбора признаков [1, 2].

В работе исследуется метод частных наименьших квадратов (PLS) [3–5]. Метод частных наименьших квадратов снижает размерности матрицы признаков и выделяет линейные комбинации признаков, которые оказывают наибольшее влияние на вектор ответов. Выделение признаков происходит итеративно, в порядке уменьшения их влияния на вектор ответов. Рассматриваются только значимые комбинации признаков, незначительно потеряв в точности прогноза. Методы PLS регрессии подробно описаны в работах [6–8]. Разницу между методом PLS и связанными с ним подходами, различные разновидности регрессии PLS можно найти в [9].

Современное состояние области и обзор нелинейных модификаций метода PLS описаны [10]. Нелинейное расширение метода PLS регрессии впервые введено в [11]. В литературе были разработаны различные модификации PLS. Предложены нелинейные методы PLS, основанные на различных моделях: сглаживающих сплайнах [12] нейронных сетях [13], базисных радиальных функциях [14], генетическом алгоритме [15].

Результатом отбора признаков является снижение размерности задачи и повышение устойчивости моделей без существенной потери точности прогноза. Методы решения задачи сравниваются на двух наборах данных, имеющих избыточную информацию. Первый набор данных — почасовые временные ряды объёмов потребления электроэнергии. Временные ряды соответствуют периоду 1999-2004 годов.

Второй набор данных взят из проекта NeuroTycho [16], в котором проектируется нейрокомпьютерный интерфейс (BCI) [17, 18] для обмена информацией между мозгом и электронным устройством. Система BCI повышает умственные и физические способности пользователя, обеспечивая прямую связь между мозгом и компьютером [19]. BCI восстанавливает поврежденную функциональность пациентов с нарушениями двигательного или когнитивного развития. Цель анализа моторных изображений заключается в распознавании предполагаемых движений по записанной активности мозга. Существуют различные методы измерения кортикальных данных для BCI описанные в [20, 21]. В работе рассматриваются сигналы электрокортикографии (ECoG) [22]. ECoG, как и другие инвазивные методы, обеспечивает более стабильную запись и лучшее разрешение в временных и пространственных областях, чем ее неинвазивные аналоги. Данные ECoG являются многомерными и измерения коррелируют как во временной, так и в пространственной областях. Задача состоит в предсказании траектории движения руки по временным рядам напряжения кортикальной активности. Сигналы ECoG измеряются через 32 канала, в то время как субъект перемещает руку.

В работах по прогнозированию сложных пространственных временных рядов прогноз выполняется поточечно. При необходимости прогнозировать несколько точек одновременно задачу предлагается решать последовательно вычисляя прогноз по точкам. При этом используются предыдущие спрогнозированные значения для получения последующих. Предлагаемый метод позволяет получать временной ряд прогнозируемых значений одновременно с учётом скрытых зависимостей не только в пространстве объектов, но и в пространстве ответов с согласованием этих двух зависимостей. Предлагаемый метод позволил существенно снизить размерность признакового пространства.

2 Постановка задачи

Задана выборка $\mathfrak{D} = (\mathbf{X}, \mathbf{Y})$, где $\mathbf{X} \in \mathbb{R}^{m\times n}$ — матрица объектов, $\mathbf{Y} \in \mathbb{R}^{m\times r}$ — матрица ответов. Способ построения выборки под определенную прикладную задачу описан в разделе *Вычислительный эксперимент*.

Предполагается, что между объектами $\mathbf{x} \in \mathbb{R}^n$ и ответами $\mathbf{y} \in \mathbb{R}^r$ существует линейная зависимость

$$\underset{1\times r}{\mathbf{y}} = \underset{1\times n}{\mathbf{x}} \cdot \underset{n\times r}{\mathbf{\Theta}} + \underset{1\times r}{\boldsymbol{\varepsilon}}, \tag{1}$$

где $\mathbf{\Theta}$ — матрица параметров модели, а $\boldsymbol{\varepsilon}$ — вектор регрессионных остатков.

Необходимо по известной выборке $\mathfrak{D}$ восстановить матрицу параметров модели (1). Оптимальные параметры находятся минимизацией функции ошибки. Введем квадратичную функцию ошибки S на выборке $\mathfrak{D}$:

$$S(\mathbf{\Theta}|\mathfrak{D}) = \Big\| \underset{m\times n}{\mathbf{X}} \cdot \underset{n\times r}{\mathbf{\Theta}} - \underset{m\times r}{\mathbf{Y}} \Big\|_2^2 = \sum_{i=1}^{m} \| \underset{1\times n}{\mathbf{x}_i} \cdot \underset{n\times r}{\mathbf{\Theta}} - \underset{1\times r}{\mathbf{y}_i} \|_2^2 \to \min_{\mathbf{\Theta}}. \tag{2}$$

Линейная зависимость столбцов матрицы $\mathbf{X}$ приводит к неустойчивому решению задачи оптимизации (2). Для устранения линейной зависимости применяются методы отбора признаков.

3 Метод частных наименьших квадратов

Для устранения линейной зависимости и снижения размерности пространства применяется метод главных компонент PCA. Основным недостатком данного метода является то, что он не учитывает взаимосвязь между объектами и ответами. Метод частных наименьших квадратов PLS проецирует матрицу объектов $\mathbf{X}$ и матрицу ответов $\mathbf{Y}$ в латентное пространство $\mathbb{R}^l$ меньшей размерности ($l < r < n$). Алгоритм PLS находит в латентном пространстве матрицу $\mathbf{T} \in \mathbb{R}^{m\times l}$, наилучшим образом описывающую исходные матрицы $\mathbf{X}$ и $\mathbf{Y}$.

Матрица объектов $\mathbf{X}$ и матрица ответов $\mathbf{Y}$ проецируются в латентное пространство следующим образом:

$$\underset{m\times n}{\mathbf{X}} = \underset{m\times l}{\mathbf{T}} \cdot \underset{l\times n}{\mathbf{P}^\mathsf{T}} + \underset{m\times n}{\mathbf{F}} = \sum_{k=1}^{l} \underset{m\times 1}{\mathbf{t}_k} \cdot \underset{1\times n}{\mathbf{p}_k^\mathsf{T}} + \underset{m\times n}{\mathbf{F}}, \tag{3}$$

$$\underset{m\times r}{\mathbf{Y}} = \underset{m\times l}{\mathbf{T}} \cdot \underset{l\times r}{\mathbf{Q}^\mathsf{T}} + \underset{m\times r}{\mathbf{E}} = \sum_{k=1}^{l} \underset{m\times 1}{\mathbf{t}_k} \cdot \underset{1\times r}{\mathbf{q}_k^\mathsf{T}} + \underset{m\times r}{\mathbf{E}}, \tag{4}$$

где $\mathbf{T}$ — матрица совместного описания объектов и ответов в латентном пространстве, причём столбцы матрицы $\mathbf{T}$ ортогональны; $\mathbf{P}$, $\mathbf{Q}$ — матрицы перехода из латентного пространства в исходные пространства; $\mathbf{E}$, $\mathbf{F}$ — матрицы невязок.

Псевдокод метода регрессии PLS приведен в алгоритме 1. Алгоритм итеративно на каждом из l шагов вычисляет по одному столбцу $\mathbf{t}_k$, $\mathbf{p}_k$, $\mathbf{q}_k$ матриц $\mathbf{T}$, $\mathbf{P}$, $\mathbf{Q}$ соответственно. После вычисления следующего набора векторов из матриц $\mathbf{X}$, $\mathbf{Y}$ вычитаются очередные одноранговые аппроксимации. Первым шагом необходимо произвести нормировку столбцов исходных матриц (вычесть среднее и разделить на стандартное отклонение). На этапе тестирования необходимо провести нормировку тестовых данных, вычислить предсказание модели 1, а затем провести обратную нормировку.

Algorithm 1 Алгоритм PLSR

Require: $\mathbf{X}, \mathbf{Y}, l$;
Ensure: $\mathbf{T}, \mathbf{P}, \mathbf{Q}$;
1: нормировать матрицы $\mathbf{X}$ и $\mathbf{Y}$ по столбцам
2: инициализировать $\mathbf{u}_0$ (первый столбец матрицы $\mathbf{Y}$)
3: $\mathbf{X}_1 = \mathbf{X}; \mathbf{Y}_1 = \mathbf{Y}$
4: **for** $k = 1, \dots, l$ **do**
5:   **repeat**
6:     $\mathbf{w}_k := \mathbf{X}_k^\mathsf{T}\mathbf{u}_{k-1}/(\mathbf{u}_{k-1}^\mathsf{T}\mathbf{u}_{k-1})$; $\quad \mathbf{w}_k := \frac{\mathbf{w}_k}{\|\mathbf{w}_k\|}$
7:     $\mathbf{t}_k := \mathbf{X}_k\mathbf{w}_k$
8:     $\mathbf{c}_k := \mathbf{Y}_k^\mathsf{T}\mathbf{t}_k/(\mathbf{t}_k^\mathsf{T}\mathbf{t}_k)$; $\quad \mathbf{c}_k := \frac{\mathbf{c}_k}{\|\mathbf{c}_k\|}$
9:     $\mathbf{u}_k := \mathbf{Y}_k\mathbf{c}_k$
10:   **until** $\mathbf{t}_k$ не стабилизируется
11:   $\mathbf{p}_k := \mathbf{X}_k^\mathsf{T}\mathbf{t}_k/(\mathbf{t}_k^\mathsf{T}\mathbf{t}_k)$, $\mathbf{q}_k := \mathbf{Y}_k^\mathsf{T}\mathbf{t}_k/(\mathbf{t}_k^\mathsf{T}\mathbf{t}_k)$
12:   $\mathbf{X}_{k+1} := \mathbf{X}_k - \mathbf{t}_k\mathbf{p}_k^\mathsf{T}$
13:   $\mathbf{Y}_{k+1} := \mathbf{Y}_k - \mathbf{t}_k\mathbf{q}_k^\mathsf{T}$

Вектора $\mathbf{t}_k$ и $\mathbf{u}_k$ из внутреннего цикла алгоритма 1 содержат информацию о матрице объектов $\mathbf{X}$ и матрице ответов $\mathbf{Y}$ соответственно. Блоки из шагов (6)-(7) и шагов (8)-(9) — аналоги алгоритма PCA для матриц $\mathbf{X}$ и $\mathbf{Y}$ [5]. Последовательное выполнение блоков позволяет учесть взаимную связь между матрицами $\mathbf{X}$ и $\mathbf{Y}$.

Теоретическое обоснование алгоритма PLS следует из следующих утверждений.

Утверждение 1. *Наилучшее описание матриц $\mathbf{X}$ и $\mathbf{Y}$ с учётом их взаимосвязи достигается при максимизации ковариации между векторами $\mathbf{t}_k$ и $\mathbf{u}_k$.*

Утверждение следует из равенства

$$\mathrm{cov}(\mathbf{t}_k, \mathbf{u}_k) = \mathrm{corr}(\mathbf{t}_k, \mathbf{u}_k) \cdot \sqrt{\mathrm{var}(\mathbf{t}_k)} \cdot \sqrt{\mathrm{var}(\mathbf{u}_k)}.$$

Максимизация дисперсий векторов $\mathbf{t}_k$ и $\mathbf{u}_k$ отвечает за сохранение информации об исходных матрицах, корреляция между векторами отвечает взаимосвязи между $\mathbf{X}$ и $\mathbf{Y}$. ■

Во внутреннем цикле алгоритма вычисляются нормированные вектора весов $\mathbf{w}_k$ и $\mathbf{c}_k$. Из данных векторов строятся матрицы весов $\mathbf{W}$ и $\mathbf{C}$ соответственно.

Утверждение 2. *В результате выполнения внутреннего цикла вектора $\mathbf{w}_k$ и $\mathbf{c}_k$ будут собственными векторами матриц $\mathbf{X}_k^\mathsf{T}\mathbf{Y}_k\mathbf{Y}_k^\mathsf{T}\mathbf{X}_k$ и $\mathbf{Y}_k^\mathsf{T}\mathbf{X}_k\mathbf{X}_k^\mathsf{T}\mathbf{Y}_k$, соответствующими максимальным собственным значениям.*

$$\mathbf{w}_k \propto \mathbf{X}_k^\mathsf{T}\mathbf{u}_{k-1} \propto \mathbf{X}_k^\mathsf{T}\mathbf{Y}_k\mathbf{c}_{k-1} \propto \mathbf{X}_k^\mathsf{T}\mathbf{Y}_k\mathbf{Y}_k^\mathsf{T}\mathbf{t}_{k-1} \propto \mathbf{X}_k^\mathsf{T}\mathbf{Y}_k\mathbf{Y}_k^\mathsf{T}\mathbf{X}_k\mathbf{w}_{k-1},$$
$$\mathbf{c}_k \propto \mathbf{Y}_k^\mathsf{T}\mathbf{t}_k \propto \mathbf{Y}_k^\mathsf{T}\mathbf{X}_k\mathbf{w}_k \propto \mathbf{Y}_k^\mathsf{T}\mathbf{X}_k\mathbf{X}_k^\mathsf{T}\mathbf{u}_{k-1} \propto \mathbf{Y}_k^\mathsf{T}\mathbf{X}_k\mathbf{X}_k^\mathsf{T}\mathbf{Y}_k\mathbf{c}_{k-1},$$

где символ $\propto$ означает равенство с точностью до мультипликативной константы.

Утверждение следует из того факта, что правила обновления векторов $\mathbf{w}_k$, $\mathbf{c}_k$ совпадают с итерацией алгоритма поиска максимального собственного значения. Данный алгоритм основан на следующем факте.

Если матрица $\mathbf{A}$ диагонализуема, $\mathbf{x}$ — некоторый вектор, то

$$\lim_{k\to\infty} \mathbf{A}^k\mathbf{x} = \lambda_{\max}(\mathbf{A}) \cdot \mathbf{v}_{\max},$$

где $\lambda_{\max}(\mathbf{A})$ — максимальное собственное значение матрицы $\mathbf{A}$, $\mathbf{v}_{\max}$ —собственный вектор матрицы $\mathbf{A}$, соответствующий $\lambda_{\max}(\mathbf{A})$. ■

Утверждение 3. *Обновление векторов по шагам (6)–(9) алгоритма 1 соответствует максимизации ковариации между векторами $\mathbf{t}_k$ и $\mathbf{u}_k$.*

Максимальная ковариация между векторами $\mathbf{t}_k$ и $\mathbf{u}_k$ равна максимальному собственному значению матрицы $\mathbf{X}_k^\mathsf{T}\mathbf{Y}_k\mathbf{Y}_k^\mathsf{T}\mathbf{X}_k$:

$$\begin{aligned}\max_{\mathbf{t}_k,\mathbf{u}_k} \mathrm{cov}(\mathbf{t}_k, \mathbf{u}_k)^2 &= \max_{\substack{\|\mathbf{w}_k\|=1\\ \|\mathbf{c}_k\|=1}} \mathrm{cov}\left(\mathbf{X}_k\mathbf{w}_k, \mathbf{Y}_k\mathbf{c}_k\right)^2 = \max_{\substack{\|\mathbf{w}_k\|=1\\ \|\mathbf{c}_k\|=1}} \mathrm{cov}\left(\mathbf{c}_k^\mathsf{T}\mathbf{Y}_k^\mathsf{T}\mathbf{X}_k\mathbf{w}_k\right)^2 = \\ &= \max_{\|\mathbf{w}_k\|=1} \mathrm{cov}\left\|\mathbf{Y}_k^\mathsf{T}\mathbf{X}_k\mathbf{w}_k\right\|^2 = \max_{\|\mathbf{w}_k\|=1} \mathbf{w}_k^\mathsf{T}\mathbf{X}_k^\mathsf{T}\mathbf{Y}_k\mathbf{Y}_k^\mathsf{T}\mathbf{X}_k\mathbf{w}_k = \\ &= \lambda_{\max}\left(\mathbf{X}_k^\mathsf{T}\mathbf{Y}_k\mathbf{Y}_k^\mathsf{T}\mathbf{X}_k\right),\end{aligned}$$

где $\lambda_{\max}(\mathbf{A})$ — максимальное собственное значение матрицы $\mathbf{A}$. Применяя утверждение 2, получаем требуемое. ■

После завершения внутреннего цикла на шаге (11) вычисляются вектора $\mathbf{p}_k$, $\mathbf{q}_k$ проецированием столбцов матриц $\mathbf{X}_k$ и $\mathbf{Y}_k$ на вектор $\mathbf{t}_k$. Для перехода на следующий шаг необходимо вычесть из матриц $\mathbf{X}_k$ и $\mathbf{Y}_k$ одноранговые аппроксимации $\mathbf{t}_k\mathbf{p}_k^\mathsf{T}$ и $\mathbf{t}_k\mathbf{q}_k^\mathsf{T}$

$$\mathbf{X}_{k+1} = \mathbf{X}_k - \mathbf{t}_k\mathbf{p}_k^\mathsf{T} = \mathbf{X} - \sum_k \mathbf{t}_k\mathbf{p}_k^\mathsf{T},$$

$$\mathbf{Y}_{k+1} = \mathbf{Y}_k - \mathbf{t}_k\mathbf{q}_k^\mathsf{T} = \mathbf{Y} - \sum_k \mathbf{t}_k\mathbf{q}_k^\mathsf{T}.$$

Тогда каждый следующий вектор $\mathbf{t}_k$ оказывается ортогонален всем векторам $\mathbf{t}_i$, $i = 1, \ldots, k$.

На Рис. 1 продемонстрирован результат работы алгоритма PLS для случая, когда размерности пространств объектов, ответов и латентного пространства равны 2 ($n = r = l = 2$). Синими и зелёными точками изображены строки матриц $\mathbf{X}$ и $\mathbf{Y}$. Точки были сгенерированы из нормального распределения с нулевым матожиданием. Красным контуром показаны линии уровня матриц ковариаций распределений. Черным изображены единичные окружности. Красные стрелки соответствуют главным компонентам. Черные стрелки соответствуют векторам матриц $\mathbf{W}$ и $\mathbf{C}$ алгоритма PLS. Вектора $\mathbf{t}_k$ и $\mathbf{u}_k$ равны проекциям матриц $\mathbf{X}_k$ и $\mathbf{Y}_k$ на вектора $\mathbf{w}_k$ и $\mathbf{c}_k$ соответственно и изображены черными плюсами. Учёт взаимной связи между матрицами $\mathbf{X}$ и $\mathbf{Y}$ отклоняет вектора $\mathbf{w}_k$ и $\mathbf{c}_k$ от направления главных компонент. Вектора $\mathbf{w}_k$ отклоняются незначительно. На первой итерации $\mathbf{c}_1$ близок к pc_1, но вектора $\mathbf{c}_k$, найденные на следующих итерациях могут оказаться сильно коррелированными. Это происходит в следствие того, что из матрицы $\mathbf{Y}$ на каждом шаге вычитается одноранговая аппроксимация, найденная в пространстве матрицы $\mathbf{X}_k$.

Для получения прогнозов модели и нахождения параметров модели домножим справа формулу (3) на матрицу $\mathbf{W}$. Строки матрицы невязок $\mathbf{E}$ ортогональны столбцам матрицы $\mathbf{W}$, поэтому

$$\mathbf{X}\mathbf{W} = \mathbf{T}\mathbf{P}^\mathsf{T}\mathbf{W}.$$

Линейное преобразование между объектами в исходном и латентном пространстве имеет вид

$$\mathbf{T} = \mathbf{X}\mathbf{W}^*, \tag{5}$$

где $\mathbf{W}^* = \mathbf{W}(\mathbf{P}^\mathsf{T}\mathbf{W})^{-1}$.

Матрица параметров модели 1 находится из уравнений (4), (5)

$$\mathbf{Y} = \mathbf{T}\mathbf{Q}^\mathsf{T} + \mathbf{E} = \mathbf{X}\mathbf{W}^*\mathbf{Q}^\mathsf{T} + \mathbf{E} = \mathbf{X}\boldsymbol{\Theta} + \mathbf{E}.$$

Таким образом, параметры модели (1) равны

$$\boldsymbol{\Theta} = \mathbf{W}(\mathbf{P}^\mathsf{T}\mathbf{W})^{-1}\mathbf{Q}^\mathsf{T}. \tag{6}$$

После нахождения параметров модели на этапе тестирования для нахождения предсказания модели необходимо

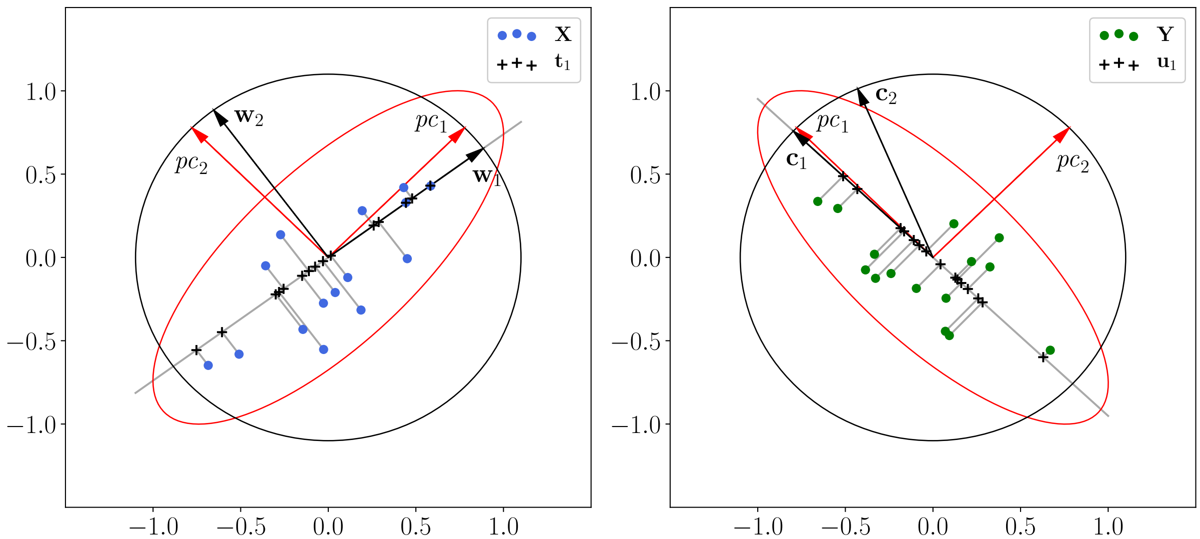

Рис. 1: Иллюстрация алгоритма PLS

- нормировать тестовые данные;
- вычислить предсказание модели с помощью линейного преобразования с матрицей $\mathbf{\Theta}$ из (6);
- провести обратную нормировку.

4 Вычислительный эксперимент

Временные ряды энергии состоят из почасовых записей (всего 52512 наблюдений), а погодные измерения проводились раз в день и содержат 2188 наблюдений. Строка матрицы $\mathbf{X}$ — локальная история сигнала за одну неделю $n = 24 \times 7$. Строка матрицы $\mathbf{Y}$ — локальный прогноз потребления электроэнергии в следующие 24 часа $r = 24$. В этом случае матрицы $\mathbf{X}$ и $\mathbf{Y}$ являются авторегрессионными матрицами(можно ссылку).

В случае данных ECoG матрица $\mathbf{X}$ состоит из пространственно-временного спектрального представления временных рядов напряжения, а матрица $\mathbf{Y}$ содержит информацию о положении руки. Процесс генерации матрицы $\mathbf{X}$ из значений напряжения описан в [23]. Признаковое описание в каждый момент времени имеет размерность 864, положение руки описывается координатами по трём осям. Пример данных напряжения с разных каналов и соответствующих пространственных координат руки представлен на Рис. 2. Для прогнозирования движения руки используется авторегрессионный подход. Один объект состоит из признакового описания в несколько отсчётов времени. Ответом является положение руки в несколько следующих моментов

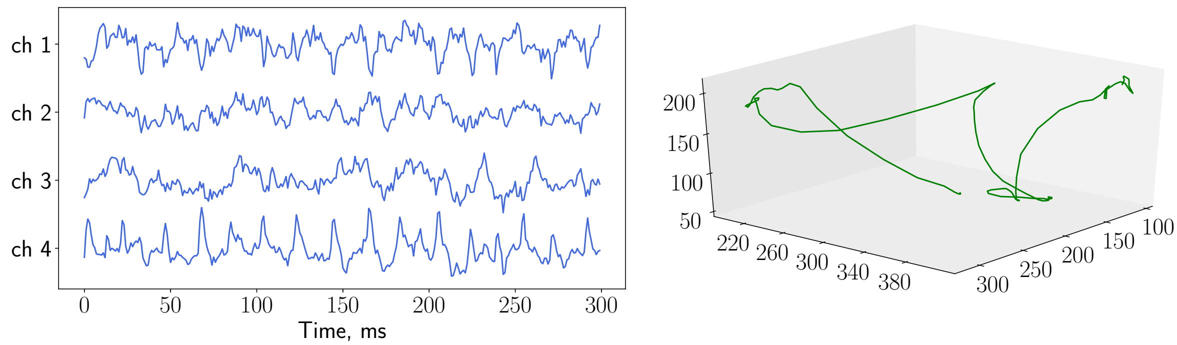

Рис. 2: Пример данных ECoG. Слева изображены данные напряжения снятые с нескольких каналов, справа — координаты руки по трём осям.

времени. Требуется предсказать положение руки в несколько следующих моментов времени.

Введём среднеквадратичную ошибку для некоторых матриц $\mathbf{A} = [a_{ij}]$ и $\mathbf{B} = [b_{ij}]$

$$\mathrm{MSE}(\mathbf{A}, \mathbf{B}) = \sum_{i,j} (a_{ij} - b_{ij})^2.$$

Для оценивания качества аппроксимации вычисляется значение нормированной среднеквадратичной ошибки

$$\mathrm{NMSE}(\mathbf{Y}, \hat{\mathbf{Y}}) = \frac{\mathrm{MSE}(\mathbf{Y}, \hat{\mathbf{Y}})}{\mathrm{MSE}(\mathbf{Y}, \bar{\mathbf{Y}})}, \tag{7}$$

где $\hat{\mathbf{Y}}$ — прогноз модели, $\bar{\mathbf{Y}}$ — константный прогноз средним значением по столбцам матрицы.

4.1 Данные потребления электроэнергии

Для нахождения оптимальной размерности l латентного пространства все данные потребления электроэнергии были разбиты на обучающую и валидационную части. Обучающая выборка состоит из 700 объектов, валидационная из 370. Зависимость нормированной квадратичной ошибки (7) от размерности l латентного пространства представлена на Рис. 3. Сначала ошибка резко падает при увеличении размерности скрытого пространства, а затем меняется незначительно.

Минимальная ошибка наблюдается при $l = 14$. Построим прогноз потребления электроэнергии при данном l. Результат аппроксимации изображен на Рис. 4. Алгоритм PLS восстановил авторегрессионную зависимость и обнаружил дневную сезонность.

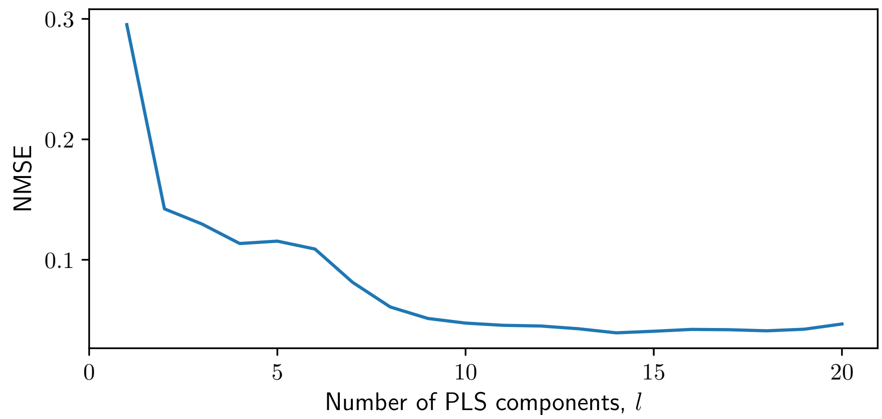

Рис. 3: Прогноз потребления электроэнергии алгоритмом PLS при размерности латентного пространства l=14

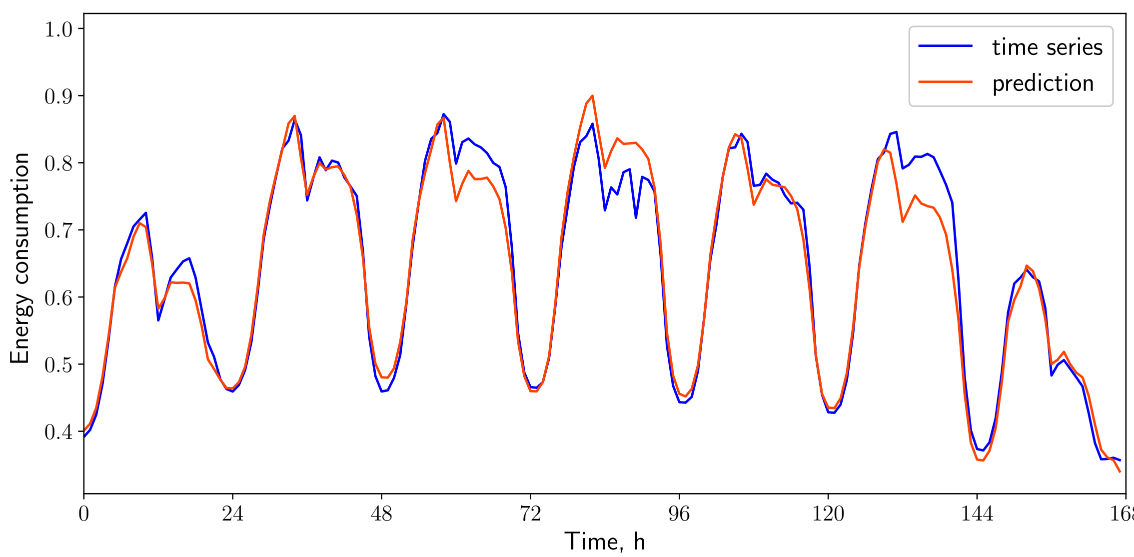

Рис. 4: Зависимость ошибки от размерности латентного пространства для данных потребления электроэнергии

4.2 Данные электрокортикограммы

На Рис. 5 представлена зависимость нормированной квадратичной ошибки (7) от размерности латентного пространства. Ошибка аппроксимации меняется незначительно при $l > 5$. Таким образом совместное описание пространственно-временного спектрального представления объектов и пространственного положения руки может быть представлено вектором размерности $l \ll n$. Зафиксируем $l = 5$. Пример аппроксимации положения руки изображен на Рис. 6. Сплошными линиями изображены истинные координаты руки по всем осям, пунктирными линиями показана аппроксимация методом PLS.

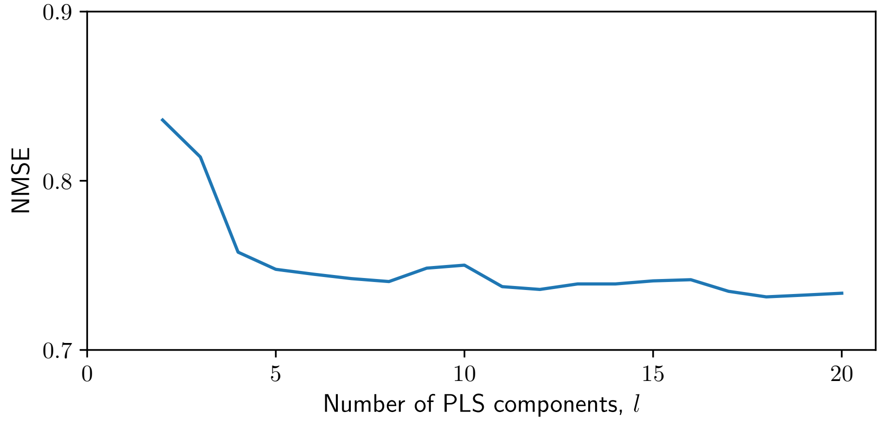

Рис. 5: Зависимость ошибки от размерности латентного пространства для данных ECoG

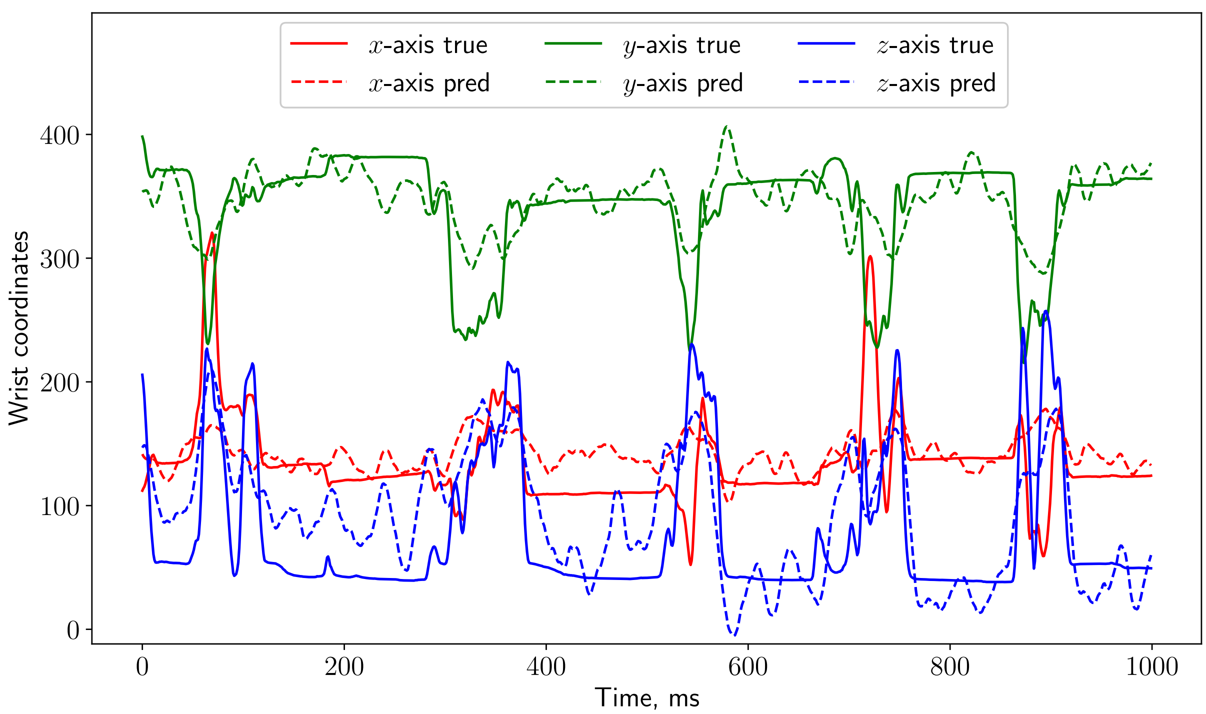

Рис. 6: Прогноз движения руки данных ECoG алгоритмом PLS при размерности латентного пространства $l = 5$

5 Заключение

В данной работе предложен новый подход к обнаружению зависимостей в пространстве зависимой переменной задачи прогнозирования временных рядов. Сравнивались результаты прогнозирования временных рядов, полученных с помощью метода частных наименьших квадратов и предложенной модификации. Проведен вычислительный эксперимент на реальных данных потребления электроэнергии в Варшаве. Построенная прогностическая модель показала высокое качество предсказания электрической нагрузки.

Список литературы

[1] AM Katrutsa and VV Strijov. Stress test procedure for feature selection algorithms. *Chemometrics and Intelligent Laboratory Systems*, 142:172–183, 2015.

[2] Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P Trevino, Jiliang Tang, and Huan Liu. Feature selection: A data perspective. *arXiv preprint arXiv:1601.07996*, 2016.

[3] Jacob A Wegelin et al. A survey of partial least squares (pls) methods, with emphasis on the two-block case. *University of Washington, Department of Statistics, Tech. Rep*, 2000.

[4] Hervé Abdi. Partial Least Squares (PLS) Regression. *Encyclopedia for research methods for the social sciences*, pages 792–795, 2003.

[5] Paul Geladi and Bruce R Kowalski. Partial least-squares regression: a tutorial. *Analytica chimica acta*, 185:1–17, 1986.

[6] Paul Geladi. Notes on the history and nature of partial least squares (PLS) modelling. *Journal of Chemometrics*, 2(January):231–246, 1988.

[7] Agnar Höskuldsson. PLS regression. *Journal of Chemometrics*, 2(August 1987):581–591, 1988.

[8] Sijmen De Jong. Simpls: an alternative approach to partial least squares regression. *Chemometrics and intelligent laboratory systems*, 18(3):251–263, 1993.

[9] Roman Rosipal and Nicole Kramer. Overview and Recent Advances in Partial Least Squares. *C. Saunders et al. (Eds.): SLSFS 2005, LNCS 3940*, pages 34–51, 2006.

[10] Roman Rosipal. Nonlinear partial least squares: An overview. *Chemoinformatics and Advanced Machine Learning Perspectives: Complex Computational Methods and Collaborative Techniques*, pages 169–189, 2011.

[11] Svante Wold, Nouna Kettaneh-Wold, and Bert Skagerberg. Nonlinear pls modeling. *Chemometrics and Intelligent Laboratory Systems*, 7(1-2):53–65, 1989.

[12] Ildiko E. Frank. A nonlinear PLS model. *Chemometrics and Intelligent Laboratory Systems*, 8(2):109–119, 1990.

[13] S Joe Qin and Thomas J McAvoy. Nonlinear pls modeling using neural networks. *Computers & Chemical Engineering*, 16(4):379–391, 1992.

[14] Xuefeng F. Yan, Dezhao Z. Chen, and Shangxu X. Hu. Chaos-genetic algorithms for optimizing the operating conditions based on RBF-PLS model. *Computers and Chemical Engineering*, 27(10):1393–1404, 2003.

[15] Hugo Hiden, Ben McKay, Mark Willis, and Gary Montague. Non-linear partial least squares using genetic. In *Genetic Programming 1998: Proceedings of the Third*, pages 128–133. Morgan Kaufmann, 1998.

[16] Project tycho http://neurotycho.org/food-tracking-task.

[17] José del R Millán, Rüdiger Rupp, Gernot Mueller-Putz, Roderick Murray-Smith, Claudio Giugliemma, Michael Tangermann, Carmen Vidaurre, Febo Cincotti, Andrea Kubler, Robert Leeb, et al. Combining brain–computer interfaces and assistive technologies: State-of-the-art and challenges. *Frontiers in Neuroscience*, 4:161, 2010.

[18] SG Mason, A Bashashati, M Fatourechi, KF Navarro, and GE Birch. A comprehensive survey of brain interface technology designs. *Annals of biomedical engineering*, 35(2):137–169, 2007.

[19] José del R Millán, Frédéric Renkens, Josep Mouriño, and Wulfram Gerstner. Brain-actuated interaction. *Artificial Intelligence*, 159(1-2):241–259, 2004.

[20] Luis Fernando Nicolas-Alonso and Jaime Gomez-Gil. Brain computer interfaces, a review. *Sensors*, 12(2):1211–1279, 2012.

[21] Setare Amiri, Reza Fazel-Rezai, and Vahid Asadpour. A review of hybrid brain-computer interface systems. *Advances in Human-Computer Interaction*, 2013:1, 2013.

[22] Andrey Eliseyev and Tetiana Aksenova. Penalized multi-way partial least squares for smooth trajectory decoding from electrocorticographic (ecog) recording. *PloS one*, 11(5):e0154878, 2016.

[23] Motrenko A. Gasanov I. Creation of approximating scalogram description in a problem of movement prediction. *Journal of Machine Learning and Data Analysis*, 3(2):160–169, 2017.