

Нелинейное снижение размерности в задачах декодирования временных рядов и прогнозирования*

Мария Владимировна, Роман Исаченко

vladimirova.maria@phystech.edu, isa-ro@yandex.ru

Московский физико-технический институт

В работе решается задача обнаружения зависимостей в прогнозируемой переменной. Используются набор гомогенных моделей, восстанавливающих прогноз по общему для всех переменных описанию объектов. Рассматривается линейная модель метода частных наименьших квадратов и ее предложенная нелинейная модификация. Находятся оптимальные параметрические преобразования исходных пространств объектов и ответов. Проводится вычислительный эксперимент на реальных данных объемов потребления электроэнергии и данных сигналов кортикограмм.

Ключевые слова: прогнозирование временных рядов; мультиколлинеарность; метод частных наименьших квадратов; PLS; нелинейный PLS

1. Введение

В работе рассматривается задача прогнозирования временных рядов в случае наличия мультиколлинеарности в данных. Методы решения данной задачи сравниваются на двух наборах данных, имеющих избыточную информацию.

Первый набор данных представляет собой временные ряды объема потребления электроэнергии в Варшаве. Электрическая энергия является важной движущей силой экономического развития, а точность прогнозов спроса является важным фактором, который ведет к успешному эффективному планированию. По этой причине энергетическим анализам необходимо руководство для лучшего выбора наиболее подходящих методов прогнозирования, чтобы обеспечить точные прогнозы тенденций потребления электроэнергии. Предполагается, что значение сигнала в данный момент времени линейно зависит от предыдущих значений этого же сигнала, поэтому данные являются мультиколлинеарными.

Второй набор данных взят из проекта Project Tycho, в котором изучалась проблема проектирования нейро-компьютерного интерфейса (BCI) для обмена информацией между мозгом и электронным устройством. Решается задача выбора функций в моделях регрессии в приложении к декодированию движения на основе электрокардиограмм (ECoG). Проблема состоит в том, чтобы предсказать траектории руки из временных рядов напряжения кортикальной активности. Описание функции каждой точки находится в пространственно-временной частотной области включает в себя сами временные ряды напряжения и их спектральные характеристики. Выбор функции имеет решающее значение для адекватного решения проблемы регрессии, поскольку электрокортикальные данные являются высокочастотными и измерения коррелируют как во временной, так и в пространственной областях.

Система BCI улучшает умственные и физические возможности пользователя, обеспечивая прямую связь между мозгом и компьютером. BCI направлены на восстановление поврежденных функциональных возможностей пациентов с механическими или когнитивными нарушениями. В данной статье предлагается новый метод выбора признаков

29 в прогнозировании движения и его реконструкции. Первый шаг к прогнозированию
30 предполагаемых движений — научиться реконструировать фактические перемещения из
31 кортикальной активности. Рассматривается проблема непрерывной реконструкции траек-
32 тории. Субдуральные сигналы ECoG измеряются через 32 или 64 канала, когда субъект
33 перемещает руку. Когда сигналы ECoG трансформируются в информационные функции,
34 проблема восстановления траектории является проблемой регрессии. Извлечение функ-
35 ции включает в себя применение некоторого спектрально-временного преобразования к
36 сигналам ECoG с каждого канала. Так как результирующее пространственно-временное
37 спектральное представление сильно избыточно, используются различные методы выбора
38 объектов и уменьшения размерности, чтобы извлечь только наиболее важные функции.

39 Для решения задачи прогнозирования используется авторегрессионная модель. Авто-
40 регрессионная модель является неустойчивой в случае наличия мультиколлинеарности в
41 исторических данных. Для решения этой проблемы необходимо используются методы от-
42 бора признаков [21], в результате чего повышается устойчивость модели без существенного
43 снижения качества прогноза.

44 В работе исследуются методы отбора признаков: метод частных наименьших квад-
45 ратов (PLS) [11] и предложенная его нелинейная модификация (cnlPLS). Метод частных
46 наименьших квадратов основан на снижении размерности матрицы признаков и выделя-
47 ет линейные комбинации признаков, которые оказывают наибольшее влияние на вектор
48 ответов. Выделение признаков происходит итеративно, в порядке уменьшения их влия-
49 ния на вектор ответов [11]. Рассматриваются только значимые комбинации признаков,
50 незначительно потеряв в точности прогноза.

51 Методы PLS регрессии подробно описаны в работах [8,9]. Разницу между методом PLS
52 и связанными с ним подходами, различные разновидности регрессии PLS можно найти
53 в [17].

54 Нелинейное расширение метода PLS регрессии впервые введено в [6]. В литературе
55 были разработаны различные модификации PLS. Предложены нелинейные методы PLS,
56 основанные на различных моделях: искусственных нейронных сетей [4], функции актива-
57 ции радиальных оснований [5], логистическая функция активации и методы оптимизации
58 роевых частиц [7], используют прямые нейронные сети [3], искусственную нейронную сеть
59 Эльмана [10].

60 Предлагается провести модификацию алгоритма PLS: совершить криволинейное и
61 нелинейное преобразования пространства целевой переменной для учета зависимостей
62 между сигналами в разные моменты времени.

63 В работе проведено сравнение двух методов отбора признаков в задаче авторегрессион-
64 ного прогнозирования сигналов (PLSR и cnlPLSR). Цель регрессии PLS [18] — предсказать
65 \mathbf{Y} по \mathbf{X} и описать их общую структуру. Когда \mathbf{Y} — вектор, а \mathbf{X} — матрица полного ран-
66 га, эта цель может быть выполнена с использованием обычной линейной регрессии. Если
67 число предикторов велико по сравнению с числом наблюдений, то \mathbf{X} будет сингулярной и
68 регрессионный подход в этом случае невозможен из-за наличия мультиколлинеарности.

69 В качестве практической проверки данных методов в ходе вычислительного экспери-
70 мента решается задача прогнозирования на реальных данных. Результатом применения
71 отбора признаков является снижение размерности задачи и повышение устойчивости мо-
72 делей без существенной потери точности прогноза.

Постановка задачи

Задана выборка $\mathcal{D} = (\mathbf{X}, \mathbf{Y})$, где $\mathbf{X} \in \mathbb{R}^{m \times n}$ — матрица объектов, \mathbf{Y} — матрица ответов. Строки матрицы \mathbf{X} являются объектами из некоторого множества $\mathbb{X} \subset \mathbb{R}^n$. Каждому объекту соответствует ответ из некоторого множества $\mathbb{Y} \subset \mathbb{R}^r$. Способ построения выборки под определенную прикладную задачу описан в разделе "Вычислительный эксперимент".

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

$$\mathbf{y} = \mathbf{x}\Theta + \epsilon, \quad (1)$$

где $\Theta \in \mathbb{R}^{n \times r}$ — матрица параметров модели, а $\epsilon \in \mathbb{R}^r$ — вектор регрессионных остатков.

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

$$S(\Theta|\mathcal{D}) = \|\mathbf{X}\Theta - \mathbf{Y}\|_2^2 = \sum_{i=1}^m \|\mathbf{x}_i\Theta - \mathbf{y}_i\|_2^2 \rightarrow \min_{\Theta}. \quad (2)$$

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

Метод частных наименьших квадратов (PLS)

Для устранения линейной зависимости и снижения размерности пространства применяется метод главных компонент. Основным недостатком данного метода является то, что он не учитывает взаимосвязь между объектами и ответами. Метод частных наименьших квадратов проецирует пространство объектов \mathbb{X} и пространство ответов \mathbb{Y} в латентное пространство \mathbb{Z} меньшей размерности. В отличие от метода главных компонент, метод частных наименьших квадратов учитывает взаимосвязь между матрицами \mathbf{X} и \mathbf{Y} и снижает размерность пространства объектов.

Матрица плана \mathbf{X} и матрица ответов \mathbf{Y} проецируются на пространство \mathbb{Z} размерности l ($l < n$) следующим образом:

$$\underset{m \times n}{\mathbf{X}} = \underset{m \times l}{\mathbf{T}} \cdot \underset{l \times n}{\mathbf{P}^T} + \underset{m \times n}{\mathbf{E}}, \quad (3)$$

$$\underset{m \times r}{\mathbf{Y}} = \underset{m \times l}{\mathbf{U}} \cdot \underset{l \times r}{\mathbf{Q}^T} + \underset{m \times r}{\mathbf{F}}, \quad (4)$$

где \mathbf{T} , \mathbf{U} — матрицы объектов и ответов в латентном пространстве \mathbb{Z} , причём $\mathbf{T}^T \mathbf{T} = \mathbf{I}_l$; \mathbf{P} , \mathbf{Q} — матрицы перехода из пространства \mathbb{Z} в исходные пространства \mathbb{X} и \mathbb{Y} ; \mathbf{E} , \mathbf{F} — матрицы невязок. Матрицы \mathbf{T} , \mathbf{U} являются матрицами объектов и ответов в спроектированном пространстве; \mathbf{P} , \mathbf{Q} — матрицы перехода из нового пространства в старое; \mathbf{E} , \mathbf{F} — матрицы невязок.

Схема алгоритма PLS изображена на Рис. 1. Необходимо найти линейное отображение из пространства \mathbb{X} в пространство \mathbb{Y} . Алгоритм PLS отображает исходные пространства в общее латентное пространство \mathbb{Z} . При этом алгоритм не только находит наилучшее приближение матриц \mathbf{X} и \mathbf{Y} , но и учитывает их взаимосвязь в новом пространстве.

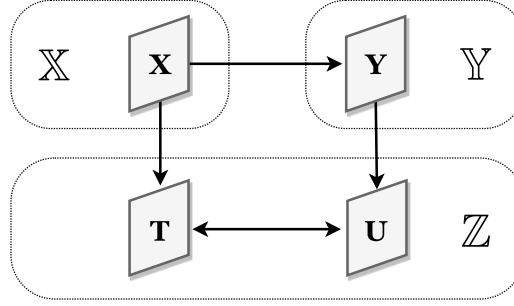


Рис. 1. Схема алгоритма PLS

Псевдокод метода регрессии PLS приведен в алгоритме 1. Алгоритм итеративно на каждом из l шагов вычисляет по одному столбцу \mathbf{t}_k , \mathbf{u}_k , \mathbf{p}_k , \mathbf{q}_k матриц \mathbf{T} , \mathbf{U} , \mathbf{P} , \mathbf{Q} соответственно. Предполагается, что вектора новых признаков \mathbf{t}_k и \mathbf{u}_k являются линейными комбинациями столбцов матриц \mathbf{X} и \mathbf{Y} соответственно.

Теоретическое обоснование алгоритма PLS следует из следующих теорем.

Теорема 1. Целью алгоритма является максимизация ковариации между векторами \mathbf{t}_k и \mathbf{u}_k . Этим достигается наилучшее описание пространств матриц \mathbf{X} и \mathbf{Y} , а также их взаимной связи

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

Во внутреннем цикле алгоритма вычисляются вектора \mathbf{w}_k и \mathbf{c}_k . Из данных векторов строятся матрицы \mathbf{W} и \mathbf{C} соответственно. Вектора \mathbf{w}_k , \mathbf{c}_k являются собственными векторами матриц $\mathbf{X}_k^\top \mathbf{Y}_k \mathbf{Y}_k^\top \mathbf{X}_k$ и $\mathbf{Y}_k^\top \mathbf{X}_k \mathbf{X}_k^\top \mathbf{Y}_k$ соответственно

$$\begin{aligned} \mathbf{w}_k &\propto \mathbf{X}_k^\top \mathbf{u}_k = \mathbf{X}_k^\top \mathbf{Y}_k \mathbf{c}_k \propto \mathbf{X}_k^\top \mathbf{Y}_k \mathbf{Y}_k^\top \mathbf{t}_k = \mathbf{X}_k^\top \mathbf{Y}_k \mathbf{Y}_k^\top \mathbf{X}_k \mathbf{w}_k, \\ \mathbf{c}_k &\propto \mathbf{Y}_k^\top \mathbf{t}_k = \mathbf{Y}_k^\top \mathbf{X}_k \mathbf{w}_k \propto \mathbf{Y}_k^\top \mathbf{X}_k \mathbf{X}_k^\top \mathbf{u}_k = \mathbf{Y}_k^\top \mathbf{X}_k \mathbf{X}_k^\top \mathbf{Y}_k \mathbf{c}_k. \end{aligned}$$

Теорема 2. Правила обновления векторов \mathbf{w}_k , \mathbf{c}_k совпадают с итерацией алгоритма поиска максимального собственного значения [20]. В результате выполнения внутреннего цикла вектора \mathbf{w}_k и \mathbf{c}_k будут являться собственными векторами матриц $\mathbf{X}_k^\top \mathbf{Y}_k \mathbf{Y}_k^\top \mathbf{X}_k$ и $\mathbf{Y}_k^\top \mathbf{X}_k \mathbf{X}_k^\top \mathbf{Y}_k$, соответствующими максимальным собственным значениям.

Теорема 3. Обновляя вектора по данным правилам, мы максимизируем ковариацию между векторами \mathbf{t}_k и \mathbf{u}_k

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

После завершения внутреннего цикла вычисляются вектора \mathbf{p}_k , \mathbf{q}_k проецированием столбцов матриц \mathbf{X}_k и \mathbf{Y}_k на вектора \mathbf{t}_k и \mathbf{u}_k . Для перехода на следующий шаг необходимо вычесть из матриц \mathbf{X}_k и \mathbf{Y}_k одноранговые аппроксимации $\mathbf{t}_k \mathbf{p}_k^\top$ и $\mathbf{t}_k \mathbf{c}_k^\top$

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

$$\mathbf{Y}_{k+1} = \mathbf{Y}_{k-1} - \mathbf{t}_k \mathbf{c}_k^\top = \mathbf{Y} - \sum_k \mathbf{t}_k \mathbf{c}_k^\top,$$

где матрицы с нижним индексом k составлены из k первых векторов, соответственно. Выражения для регрессии матрицы \mathbf{X} и матрицы \mathbf{Y} на полученные ортогональные вектора $\mathbf{t}_1, \dots, \mathbf{t}_k$

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

$$\mathbf{Y} = \sum_k \mathbf{t}_k \mathbf{c}_k^\top + \mathbf{Y}_{k+1} = \mathbf{T}_k \mathbf{C}_k^\top + \mathbf{Y}_{k+1}.$$

122 Для перехода из пространства \mathbf{X} в пространство \mathbf{T} применяется линейное преобразо-
123 вание

$$\mathbf{T} = \mathbf{XW}(\mathbf{P}^\top \mathbf{W})^{-1}.$$

Для нахождения коэффициентов регрессии \mathbf{B} преобразуем полученное за l шагов алгоритма выражение для регрессии матрицы \mathbf{Y} в предположении, что \mathbf{Y}_{l+1} является матрицей невязки,

$$\mathbf{Y} = \mathbf{TC}^\top + \mathbf{Y}_{l+1} = \mathbf{XW}(\mathbf{P}^\top \mathbf{W})^{-1} \mathbf{C}^\top + \mathbf{Y}_{l+1} = \mathbf{XB} + \mathbf{Y}_{l+1}.$$

Таким образом,

$$\mathbf{B} = \mathbf{W}(\mathbf{P}^\top \mathbf{W})^{-1} \mathbf{C}^\top.$$

Алгоритм 1 Алгоритм PLSR

Вход: $\mathbf{X}, \mathbf{Y}, l$;

Выход: $\mathbf{T}, \mathbf{U}, \mathbf{P}, \mathbf{Q}$;

- 1: нормировать матрицы \mathbf{X} и \mathbf{Y}
 - 2: инициализировать \mathbf{u}_1 (первый столбец матрицы \mathbf{Y})
 - 3: $\mathbf{X}_1 = \mathbf{X}; \mathbf{Y}_1 = \mathbf{Y}$
 - 4: для $k = 1, \dots, l$
 - 5: повторять
 - 6: $\mathbf{w}_k := \mathbf{X}_k^\top \mathbf{u}_k / (\mathbf{u}_k^\top \mathbf{u}_k); \quad \mathbf{w}_k := \frac{\mathbf{w}_k}{\|\mathbf{w}_k\|}$
 - 7: $\mathbf{t}_k := \mathbf{X}_k \mathbf{w}_k$
 - 8: $\mathbf{c}_k := \mathbf{Y}_k^\top \mathbf{t}_k / (\mathbf{t}_k^\top \mathbf{t}_k)$
 - 9: $\mathbf{u}_k := \mathbf{Y}_k \mathbf{c}_k / (\mathbf{c}_k^\top \mathbf{c}_k)$
 - 10: пока \mathbf{t}_k не стабилизируется
 - 11: $\mathbf{p}_k := \mathbf{X}_k^\top \mathbf{t}_k / (\mathbf{t}_k^\top \mathbf{t}_k), \quad \mathbf{q}_k := \mathbf{Y}_k^\top \mathbf{u}_k / (\mathbf{u}_k^\top \mathbf{u}_k)$
 - 12: $\mathbf{X}_{k+1} := \mathbf{X}_k - \mathbf{t}_k \cdot \left(\frac{\mathbf{X}_k^\top \mathbf{t}_k}{\mathbf{t}_k^\top \mathbf{t}_k} \right)^\top$
 - 13: $\mathbf{Y}_{k+1} := \mathbf{Y}_k - \mathbf{t}_k \cdot \left(\frac{\mathbf{Y}_k^\top \mathbf{t}_k}{\mathbf{t}_k^\top \mathbf{t}_k} \right)^\top$
-

Модификация метода частных наименьших квадратов (cnlPLS)

Предлагается провести модификацию алгоритма PLS: совершить криволинейное и нелинейное преобразования пространства целевой переменной и независимой переменной для учета мультиколлинеарности между значениями сигнала в различные моменты времени. Схема модифицированного алгоритма представлена на рис. 2. После применения к исходным матрицам \mathbf{X} и \mathbf{Y} нелинейных преобразований F_x и F_y соответственно используется линейный алгоритм метода частных наименьших квадратов.

$$\begin{aligned} \mathbf{X} &\xrightarrow{F_x(\mathbf{X}, \mathbf{v}_x)} \tilde{\mathbf{X}} = \mathbf{T} \times \mathbf{P}^\top + \mathbf{E} \\ \mathbf{Y} &\xrightarrow{F_y(\mathbf{Y}, \mathbf{v}_y)} \tilde{\mathbf{Y}} = \mathbf{U} \times \mathbf{Q}^\top + \mathbf{F} \end{aligned}$$

Рис. 2. Размерности векторов в алгоритме cnlPLS

Нелинейные преобразования

Рассматриваются нелинейное параметрическое преобразование пространства зависимой переменной \mathbf{Y} и независимой переменной \mathbf{X} (примеры преобразований представлены в табл. 1). Преобразование и вектор параметров, относящиеся к зависимой переменной и независимой переменной, обозначим соответственно $F_y(\mathbf{Y}, \mathbf{v}_y)$ и $F_x(\mathbf{X}, \mathbf{v}_x)$ и введем переменные для преобразованных пространств

$$\tilde{\mathbf{Y}} = F_y(\mathbf{Y}, \mathbf{v}_y), \quad \tilde{\mathbf{X}} = F_x(\mathbf{X}, \mathbf{v}_x). \quad (5)$$

Функции для криволинейных преобразований удовлетворяют следующим условиям:

- $F : \mathbb{R} \rightarrow \mathbb{R}$,
- $F(0) = 0$,
- F дифференцируется по параметрам \mathbf{v}_y ,
- существует F^{-1} .

№	Функция	Параметры
1	$F(x) = \text{sign}(x) \exp(a)(\exp(b x) - 1)$	$a, b > 0$
2	$F(x) = \text{sign}(x) \exp(a)(\exp(b \ln(1 + x)) - 1)$	$a, b > 0$
3	$F(x) = \text{sign}(x) \exp(a)(\exp(b x ^{1/2}) - 1)$	$a, b > 0$
4	$F(x) = \text{sign}(x) \exp(a)(\exp(b x ^{1/3}) - 1)$	$a, b > 0$
5	$F(x) = \text{sign}(x) \exp(a)(\exp(b x ^{1/4}) - 1)$	$a, b > 0$
6	$F(x) = \text{sign}(x) \exp(a)(\exp(b x ^2) - 1)$	$a, b > 0$

Таблица 1. Нелинейные преобразования

Для обучения параметров \mathbf{v}_y используется градиентный метод. Предлагается подход для обновления весов \mathbf{v}_y , основанный на линеаризации функции преобразования. Разложим (5) в ряд Тейлора до второго порядка:

$$\mathbf{u} \approx \mathbf{u}_0 + \frac{\partial \mathbf{u}}{\partial \mathbf{v}_y} \Delta \mathbf{v}_y.$$

Для вычисления $\Delta \mathbf{v}_y$ предложены следующие шаги. Рассматривается разница $\mathbf{u} - \mathbf{u}_0 = \frac{\partial \mathbf{u}}{\partial \mathbf{v}_y} \Delta \mathbf{v}_y$. Определется рассогласование

$$\mathbf{u} - \mathbf{u}_0 \approx \frac{\partial \mathbf{u}}{\partial \mathbf{v}_y} \Delta \mathbf{v}_y = \mathbf{J}_u \Delta \mathbf{v}_y,$$

143 где матрица \mathbf{J}_u состоит из частных производных $\left\{ \frac{\partial \mathbf{u}}{\partial \mathbf{v}_y} \right\}$, вычисленных при известном зна-
144 чении переменной \mathbf{u} :

$$\mathbf{J}_u = \frac{\partial \mathbf{u}}{\partial \mathbf{v}_y} = \frac{\partial}{\partial \mathbf{v}_y} (\tilde{\mathbf{Y}} \mathbf{c}) = \frac{1}{(\mathbf{t}^\top \mathbf{t})} \frac{\partial}{\partial \mathbf{v}_y} (\tilde{\mathbf{Y}} \tilde{\mathbf{Y}}^\top \mathbf{t}) = \frac{1}{(\mathbf{t}^\top \mathbf{t})} \left(\frac{\partial \tilde{\mathbf{Y}}}{\partial \mathbf{v}_y} \cdot \tilde{\mathbf{Y}}^\top + \tilde{\mathbf{Y}} \cdot \frac{\partial \tilde{\mathbf{Y}}^\top}{\partial \mathbf{v}_y} \right) \mathbf{t}.$$

145 Правило обновления для вектора $\Delta \mathbf{v}$ является решением задачи регрессии рассогла-
146 сования

$$\Delta \mathbf{v}_y = (\mathbf{J}_u^\top \mathbf{J}_u)^{-1} \mathbf{J}_u^\top (\mathbf{u} - \mathbf{u}_0). \quad (6)$$

147 Аналогично преобразованию зависимой переменной сводим задачу обновления вектора
148 параметров \mathbf{v}_x к задаче линейной регрессии:

$$\begin{aligned} \mathbf{t} - \mathbf{t}_0 &\approx \frac{\partial \mathbf{t}}{\partial \mathbf{v}_x} \Delta \mathbf{v}_x = \mathbf{J}_t \Delta \mathbf{v}_x \\ \Delta \mathbf{v}_x &= (\mathbf{J}_t^\top \mathbf{J}_t)^{-1} \mathbf{J}_t^\top (\mathbf{t} - \mathbf{t}_0). \end{aligned}$$

149 Преобразование независимой переменной

150 Алгоритм cnlPLSR

151 В данном разделе представлен модифицированный метод PLSR, содержащий шаги
152 преобразования целевой переменной. Аналогично методу PLSR (алгоритм 1), алгоритм
153 ?? начинается с инициализации вектора \mathbf{u} , а обновления весов преобразования считается
154 с помощью рассогласования \mathbf{e} для вектора \mathbf{u} , вычисленного в цикле и на предыдущей
155 итерации.

Алгоритм 2 Алгоритм snlPLSR с преобразованием пространства объектов 2

Вход: $\mathbf{X}, \mathbf{Y}, l$;

Выход: $\mathbf{T}, \mathbf{U}, \mathbf{P}, \mathbf{Q}, \mathbf{v}_x, \mathbf{v}_y$;

```

1: инициализировать  $\mathbf{v}_x$  и  $\mathbf{v}_y$ 
2:  $\mathbf{T} := \mathbf{U} := \mathbf{P} := \mathbf{Q} := \emptyset$ 
3: для  $k = 1, \dots, l$ 
4:   инициализировать  $\mathbf{t}$  (первый столбец матрицы  $\mathbf{X}$ )
5:   инициализировать  $\mathbf{u}$  (первый столбец матрицы  $\mathbf{Y}$ )
6:   повторять
7:      $\mathbf{t}_0 := \mathbf{t}, \mathbf{u}_0 = \mathbf{u}$ 
8:      $\tilde{\mathbf{X}} := F_x(\mathbf{X}, \mathbf{v}_x); \quad \tilde{\mathbf{Y}} = F_y(\mathbf{Y}, \mathbf{v}_y)$ 
9:      $\mathbf{w} := \tilde{\mathbf{X}}^\top \mathbf{u} / (\mathbf{u}^\top \mathbf{u}); \quad \mathbf{w} := \frac{\mathbf{w}}{\|\mathbf{w}\|}$ 
10:     $\mathbf{t} := \tilde{\mathbf{X}} \mathbf{w}$ 
11:     $\Delta \mathbf{v}_x = (\mathbf{J}_t^\top \mathbf{J}_t)^{-1} \mathbf{J}_t^\top (\mathbf{t} - \mathbf{t}_0)$ , где  $\mathbf{J}_t := \frac{\partial \mathbf{t}}{\partial \mathbf{v}_x}$ 
12:     $\mathbf{v}_x := \mathbf{v}_x + \Delta \mathbf{v}_x$ 
13:     $\mathbf{c} := \tilde{\mathbf{Y}}^\top \mathbf{t} / (\mathbf{t}^\top \mathbf{t}); \quad \mathbf{c} := \frac{\mathbf{c}}{\|\mathbf{c}\|}$ 
14:     $\mathbf{u} := \tilde{\mathbf{Y}} \mathbf{c}$ 
15:     $\Delta \mathbf{v}_y = (\mathbf{J}_u^\top \mathbf{J}_u)^{-1} \mathbf{J}_u^\top (\mathbf{u} - \mathbf{u}_0)$ , где  $\mathbf{J}_u := \frac{\partial \mathbf{u}}{\partial \mathbf{v}_y}$ 
16:     $\mathbf{v}_y := \mathbf{v}_y + \Delta \mathbf{v}_y$ 
17:   пока  $\mathbf{t}$  не стабилизируется
18:    $\mathbf{T} := \text{concat}[\mathbf{T}; \mathbf{t}]; \mathbf{U} := \text{concat}[\mathbf{U}; \mathbf{u}]$ 
19:    $\mathbf{p} := \tilde{\mathbf{X}}^\top \mathbf{t} / (\mathbf{t}^\top \mathbf{t}), \mathbf{q} := \tilde{\mathbf{Y}}^\top \mathbf{u} / (\mathbf{u}^\top \mathbf{u})$ 
20:    $\mathbf{P} := \text{concat}[\mathbf{P}; \mathbf{p}]; \mathbf{Q} := \text{concat}[\mathbf{Q}; \mathbf{q}]$ 
21:    $\tilde{\mathbf{X}} := \tilde{\mathbf{X}} - \mathbf{t} \mathbf{p}^\top$ 
22:    $\tilde{\mathbf{Y}} := \tilde{\mathbf{Y}} - \mathbf{t} \mathbf{c}^\top$ 
23:    $\mathbf{X} = F_x^{-1}(\tilde{\mathbf{X}}, \mathbf{v}_x); \mathbf{Y} = F_y^{-1}(\tilde{\mathbf{Y}}, \mathbf{v}_y)$ 

```

Вычислительный эксперимент

В случае данных электроэнергии i -ая строка матрицы \mathbf{X} — локальная история сигнала (n значений сигнала, начиная с момента i), а i -ая строка матрицы \mathbf{Y} — локальный прогноз, то есть r значений сигнала, начиная с момента $n + 1$. 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 (ссылка на Мотренко).

В рамках вычислительного эксперимента строится прогноз временных рядов. В ходе эксперимента сравниваются методы PLSR, нелинейных автоэнкодеров и snlPLS. Сравнение проводится на реальных данных объемов потребления электроэнергии в Польше.

Вычислительный эксперимент, продемонстрированный в этом разделе, основан на данных электроэнергии. Данные состоят из временного ряда польских электрических нагрузок и временных рядов погоды в Варшаве (долгота: 21,25, широта: 52,30, высота над уровнем моря: 94). Временные ряды энергии состоят из почасовых записей (всего 52512 наблюдений), а погодные измерения проводились раз в день и содержат 2188 наблюдений. Многомасштабные временные ряды соответствуют периоду 1999-2004 годов. Результаты,

Алгоритм	N=3	N=5	N=10	N=20
PLS	0,00404	0,00337	0,00151	0,00135
cnlPLS $g(x) = \text{sign}(x) \exp(a)(\exp(b x) - 1)$	0.00529	0.00514	0.00536	0.00506
cnlPLS $g(x) = \text{sign}(x) \exp(a)(\exp(b \ln(1 + x) - 1)$	0.00362	0.00386	0.00326	0.00317
cnlPLS $g(x) = \text{sign}(x) \exp(a)(\exp(b x ^{1/2}) - 1)$	0.00272	0.00236	0.00287	0.00128
cnlPLS $g(x) = \text{sign}(x) \exp(a)(\exp(b x ^{1/3}) - 1)$	0.00241	0.00233	0.00221	0.00173
cnlPLS $g(x) = \text{sign}(x) \exp(a)(\exp(b x ^{1/4}) - 1)$	0.00796	0.00768	0.00737	0.00803
cnlPLS $g(x) = \text{sign}(x) \exp(a)(\exp(b x ^2) - 1)$	0.00816	0.00798	0.00796	0.00775

Таблица 2. Значения ошибки MSE для разных чисел компонент и разных функций

полученные с этим набором данных, являются иллюстрацией предлагаемых методов, поскольку данные содержат многомасштабные временные ряды, имеющие различный характер.

Примеры работы алгоритма приведены на рис. ???. Метод успешно делает краткосрочный прогноз (до 10 дней). С увеличением горизонта прогнозирования предсказание смещается.

Результаты вычислительного эксперимента для предложенного модифицированного алгоритма cnlPLS представлены на рис. ??. На графиках изображены сглаженные зависимости ошибки MSE от числа компонент в алгоритме для разных функций. Из графиков видно, что для функций (a) – (e) ошибка при увеличении числа компонент падает, затем колеблется, слабо меняясь. Ошибка алгоритма с функцией (f) увеличивается при увеличении числа компонент. Это означает, что преобразование, выполненное в пространстве целевой переменной с помощью функции (f), плохо описывает зависимость. Меньшую ошибку имеют функции, растущие медленнее, а именно (d) и (e).

В табл. 2 продемонстрировано увеличение точности прогнозирования при использовании криволинейного преобразования в пространстве зависимой переменной, но увеличение точности в пределах погрешности алгоритма (0.0005-0.0010). Функции с быстрым ростом не позволяют описать зависимость.

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.

Литература

- [1] *Thrun, Sebastian and Pratt, Lorien* Learning to learn // Springer Science & Business Media, 2012.
- [2] *Chong, Il Gyo and Jun, Chi Hyuck* Performance of some variable selection methods when multicollinearity is present // *Chemometrics and Intelligent Laboratory Systems*, 2005. Vol. 78. No. 1. P. 103–112.
- [3] *Xuefeng, Yan* Hybrid artificial neural network based on BP-PLSR and its application in development of soft sensors // *Chemometrics and Intelligent Laboratory Systems*, 2010. Vol. 103. No. 2. P. 152–159.
- [4] *Mcavovt, J. and Process, Chemical Title* // *Journal name*, 2005. Vol. 16. No. 4. P. 379–391.
- [5] *Yan, Xuefeng F. and Chen, Dezhao Z. and Hu, Shangru X.* Chaos-genetic algorithms for optimizing the operating conditions based on RBF-PLS model // *Computers and Chemical Engineering*, 2003. Vol. 27. No. 10. P. 1393–1404.
- [6] *Frank, Ildiko E.* A nonlinear PLS model // *Chemometrics and Intelligent Laboratory Systems*, 1990. Vol. 8. No. 2. P. 109–119.
- [7] *Zhou, Yan Ping and Jiang, Jian Hui and Lin, Wei Qi and Xu, Lu and Wu, Hai Long and Shen, Guo Li and Yu, Ru Qin* Artificial neural network-based transformation for nonlinear partial least-square regression with application to QSAR studies // *Talanta*, 2007. Vol. 71. No. 2. P. 848–853.
- [8] *Chong, Il Gyo and Jun, Chi Hyuck* Notes on the history and nature of partial least squares (PLS) modelling // *Journal of Chemometrics*, 1988. Vol. 2. No. January. P. 231–246.
- [9] *Höskuldsson, Agnar* PLS regression // *Chemometrics and Intelligent Laboratory Systems*, 1987. Vol. 2. No. August. P. 581–591.
- [10] *Bulut, Elif and Egrioglu, Erol* A New Partial Least Square Method Based on Elman Neural Network // *Chemometrics and Intelligent Laboratory Systems*, 2005. Vol. 4. No. 4. P. 154–158.
- [11] *Ng, Kee Siong* A Simple Explanation of Partial Least Squares // *Journal title*, 2013. Vol. volume. No. number. P. 1–10.
- [12] *Rosipal, Roman* Nonlinear partial least squares: An overview // *Chemoinformatics and Advanced Machine Learning Perspectives: Complex Computational Methods and Collaborative Techniques*, 2011. Vol. number. No. number. P. 169–189.
- [13] *Wold, Svante and Kettaneh-Wold, Nouna and Skagerberg, Bert* Nonlinear PLS modeling // *Chemometrics and Intelligent Laboratory Systems*, 1989. Vol. 7. No. 1-2. P. 53–65.
- [14] *Rosipal, Roman and Kramer, Nicole* Overview and Recent Advances in Partial Least Squares // *????? C. Saunders et al. (Eds.): SLSFS 2005, LNCS 3940*, 2006. Vol. ?. No. ?. P. 34–51.
- [15] *Lu, Wen-Cong and Chen, Nian-Yi and Li, Guo-Zheng and Yang, Jie* Multitask Learning Using Partial Least Squares Method // *Proceedings of the Seventh International Conference on Information Fusion; International Society of Information Fusion*, 2004. Vol. 1. P. 79–84.
- [16] *Varnek, Alexandre and Baskin, Igor* Machine learning methods for property prediction in chemoinformatics: Quo Vadis? // *Journal of Chemical Information and Modeling*, 2012. Vol. 52. No. 6. P. 1413–1437.
- [17] *Lehky, Sidney R. and Kiani, Roozbeh and Esteky, Hossein and Tanaka, Keiji* Dimensionality of object representations in monkey inferotemporal cortex // *Neural computation*, 2014. Vol. 1872. No. 10. P. 1840–1872.
- [18] *Abdi, Hervé* Partial Least Squares (PLS) Regression // *Encyclopedia for research methods for the social sciences*, 2003. P. 792–795.
- [19] *Caruana, Rich and de Sa, Virginia R.* Benefitting from the Variables that Variable Selection Discards // *Journal of Machine Learning Research*, 2003. Vol. 3. No. 7-8. P. 1245–1264.

- 242 [20] *Mises R. V., Pollaczek-Geiringer H.* Praktische Verfahren der Gleichungsauflösung // ZAMM-
243 Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und
244 Mechanik, 1929. Vol. 9. No. 1. P. 58–77.
- 245 [21] *Li J. et al.* Feature selection: A data perspective // arXiv preprint arXiv:1601.07996, 2016.