

МОСКОВСКИЙ ФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ИНСТИТУТ)

На правах рукописи

Гончаров Алексей Владимирович

МЕТОДЫ МНОЖЕСТВЕННОГО ВЫРАВНИВАНИЯ
ПРОСТРАНСТВЕННО-ВРЕМЕННЫХ ОБЪЕКТОВ

05.13.17 — Теоретические основы информатики

Диссертация на соискание учёной степени
кандидата физико-математических наук

Научный руководитель:
д.ф.-м.н. В. В. Стрижов

Москва — 2019

Оглавление

| | Стр. |
|--|--------|
| Введение | 4 |
| Список сокращений и условных обозначений | 10 |
| Список иллюстраций | 12 |
| Список таблиц | 14 |
| Глава 1. Обзор литературы | 15 |
| Глава 2. Обобщение множественного выравнивания и построение центроида | 23 |
| 2.1. Множественное выравнивание временных рядов | 23 |
| 2.1.1. Базовый алгоритм вычисления расстояния методом DTW | 26 |
| 2.1.2. Общий алгоритм вычисления расстояния методом DTW | 27 |
| 2.2. Постановка задачи выбора функции расстояния с точки зрения классов эквивалентности. | 28 |
| 2.3. Постановка задачи построения центроида | 29 |
| 2.3.1. Решение задачи нахождения центроида методом DBA. | 30 |
| 2.4. Постановки задач, решаемых метрическими методами | 30 |
| 2.4.1. Задача классификации | 31 |
| 2.4.2. Задача кластеризации | 31 |
| 2.4.3. Задача поиска подпоследовательности | 32 |
| Глава 3. Обобщение матричного, многомерного и единовременного выравнивания временных рядов | 34 |
| 3.1. Матричное выравнивание | 34 |
| 3.1.1. Постановка задачи матричного выравнивания | 34 |
| 3.1.2. Алгоритм вычисления значения расстояния (3.3). | 36 |
| 3.2. Выравнивание многомерных временных рядов | 37 |
| 3.2.1. Постановка задачи многомерного выравнивания | 37 |
| 3.2.2. Алгоритм вычисления многомерного расстояния DTW | 39 |
| 3.3. Одновременное выравнивание нескольких временных рядов | 39 |
| 3.3.1. Алгоритм единовременного вычисления расстояния DTW | 40 |
| Глава 4. Обобщение метода выравнивания в непрерывном пространстве | 41 |
| 4.1. Непрерывное выравнивание временных рядов | 41 |
| 4.2. Переход от дискретных объектов к непрерывным | 41 |
| 4.3. Расстояние DTW на непрерывной оси времени | 42 |
| 4.4. Выравнивающий путь и его свойства в случае непрерывной оси времени | 42 |
| 4.5. Вычисление выравнивающего пути | 44 |
| Глава 5. Взвешенное выравнивание, анализ отклонений и выравнивание скрытых представлений как следствия обобщения метода выравнивания | 45 |

| | |
|---|--------|
| 5.1. Взвешенное выравнивание | 45 |
| 5.1.1. Постановка задачи поиска оптимального метрического классификатора | 45 |
| 5.1.2. Вычисление значения функции расстояния | 46 |
| 5.1.3. Вычисление параметров модели классификатора | 47 |
| 5.1.4. Постановка задачи построения центроида | 48 |
| 5.2. Свойства множества отклонений выравнивающего пути | 51 |
| 5.2.1. Постановка задачи поиска оптимального классификатора и функции расстояния | 51 |
| 5.2.2. Анализ отклонений | 52 |
| 5.3. Выравнивание скрытых представлений (TBD) | 53 |
| Глава 6. Анализ методов метрической классификации путем постановки вычислительных экспериментов | 55 |
| 6.1. Вычислительный эксперимент по метрической классификации методом ближайшего соседа (раздел 2.2) | 55 |
| 6.2. Вычислительный эксперимент по классификации синтетических и реальных временных рядов (раздел 2.4) | 60 |
| 6.3. Вычислительный эксперимент по классификации временных рядов методом ближайшего центроида (раздел 2.6) | 65 |
| 6.4. Вычислительный эксперимент по многоклассовой классификации временных рядов методами Random forest, SVM, KNN (раздел 2.7) | 68 |
| 6.5. Вычислительный эксперимент по метрической многоклассовой классификации и классификации в пространстве временных рядов (магистерская диссертация Моргачева) | 72 |
| 6.6. Вычислительный эксперимент, подтверждающий корректность применения метода DTW к пространственно-временным объектам | 76 |
| 6.7. Вычислительный эксперимент по подбору оптимальной функции расстояния для кластеризации многомерных временных рядов | 83 |
| Глава 7. Заключение | 87 |
| Список литературы | 88 |

Введение

Актуальность темы. Исследование направлено на решение проблемы построения моделей классификации и кластеризации пространственно-временных объектов в условиях недостаточного объема данных для обучения моделей. Следствием недостаточного объема данных в задачах классификации и кластеризации временных рядов является неустойчивость моделей к локальным и глобальным сдвигам, сжатиям и растяжениям пространственно-временных объектов по оси времени. Исследуются метрические методы, свойства которых позволяют им оставаться устойчивыми к наличию подобных искажений и быть применимыми для работы с маленькими выборками и отсутствием априорной информации об объектах.

На защиту выносятся:

1. Исследованы способы представления пространственно-временных рядов и выравнивающего пространства;
2. Предложен метод выравнивания пространственно-временных рядов в их непрерывном представлении;
3. Предложен метод нахождения центроида при кластеризации (нахождении класса эквивалентности) пространственно-временных рядов;
4. Исследованы свойства метрических пространств, задаваемых выравнивающим отображением;
5. Предложены методы взвешенного выравнивания и выравнивания скрытых представлений;
6. Предложенные методы проанализированы с помощью вычислительных экспериментов.

Цели работы. Объектом исследования работы выбран пространственно-временной ряд и его непрерывные аппроксимирующие представления. В работе решается проблема нетривиальности построения классов (кластеров) эквивалентности в связи с необходимостью использования выравнивающих монотонных преобразований. Используется непрерывное представление исследуемого объекта и выравнивающего отображения, методы функционального анализа данных и дифференциальной геометрии. Цели работы:

1. Предложить метод множественного выравнивания пространственно-временных рядов для решения задачи вычисления расстояния для набора временных рядов, многомерных временных рядов и временных рядов с несколькими осями времени;
2. Предложить метод взвешенного выравнивания временных рядов для выделения наиболее значимых для решения задачи классификации участков временного ряда;

3. Предложить метод выравнивания скрытых представлений для решения задачи иерархического выравнивания временного ряда на разных уровнях абстракции и для снижения вычислительной сложности при решении задачи;
4. Предложить метод выравнивания непрерывных объектов для решения задачи вычисления расстояния между объектами в пространстве непрерывного времени;
5. Разработать алгоритм построения моделей и провести вычислительный эксперимент для сравнения различных методов выравнивания и вычисления значения расстояния по множеству невязок при решении задачи метрической классификации и кластеризации.

Методы исследования. Решается задача расчета расстояния между двумя пространственно-временными объектами, которые могут быть нелинейно деформированы вдоль оси времени. Применяется метод динамического выравнивания временных рядов: рассчитывается матрица попарных расстояний, внутри которой при помощи методов динамического программирования осуществляется поиск выравнивающего пути между двумя временными рядами. Этот путь используется для расчета расстояния, так как выравнивает оси времени пространственно-временных объектов друг относительно друга оптимальным образом.

На 1 справа показана матрица парных расстояний для случая $\rho(x, y) \in [0, 1]$, длина объектов $m = n = 8$. На ?? показана матрица стоимости Ω алгоритма оптимального выравнивания. Путь наименьшей стоимости показан точками. Его начало и конец фиксированы в элементах с индексами $(0, 0)$ и (n, m) .

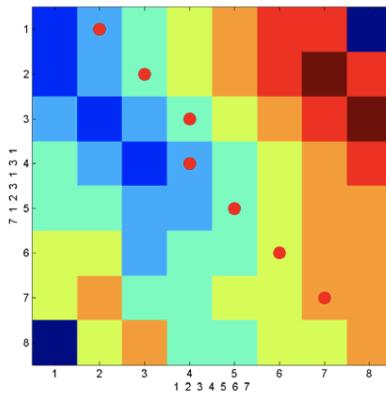


Рис. 1. Матрица стоимости оптимального выравнивания

Так как в общем случае измерений у объекта может быть много, а аналогов временных шкал может быть также несколько - необходимо подготовить обобщение метода для выравнивания произвольного пространственно-временного объекта. В работе исследуются методы множественного выравнивания между объектами такой сложной структуры.

Так как в общем случае могут сравниваться объекты с разной частотой дискретизации или же со случайным порядком детектируемых событий, необходимо предложить абстракцию метода, которая будет устойчива к подобным условиям. Таким образом, работа предлагает провести переход к непрерывному представлению объектов и работать с ними как с непрерывными, а не дискретными объектами.

Так как существует потребность расчета расстояния между сложными объектами (например, видеопоследовательности), предлагается провести адаптацию метода множественного выравнивания к работе со скрытыми представлениями.

Так как при анализе множества отклонений теряется информация о степени важности различных измерений временного ряда, а также информация о специфичных расхождениях двух временных рядов друг относительно друга, предлагается дополнительно рассмотреть свойства пространства отклонений и провести адаптацию метода для учета этой информации: вводятся понятия взвешенного выравнивания и многомерной функцией расстояния на основе множества отклонений.

Основные положения, выносимые на защиту.

1. Исследованы способы представления пространственно-временных рядов и выравнивающего пространства. Предложен метод множественно выравнивания пространственно-временных рядов, выравнивающий несколько временных рядов, временные ряды с несколькими осями времени и многомерные временные ряды. Применимость метода показана на частных случаях таких объектов;
2. Предложен метод выравнивания пространственно-временных объектов в их непрерывном представлении. Метод проводит выравнивание временных рядов как непрерывных функций от времени;
3. Предложен метод нахождения центроида при кластеризации (нахождении класса эквивалентности) пространственно-временных рядов;
4. Исследованы свойства метрических пространств, задаваемых выравнивающими отображением;
5. Предложены методы взвешенного выравнивания и выравнивания скрытых представлений. Методы позволяют повысить точность и получить дополнительную информацию об отношениях выравниваемых объектов, а также оперировать объектами более сложной структуры;
6. Предложенные методы проанализированы с помощью вычислительных экспериментов. Приведены рекомендации по практическому использованию разработанных алгоритмов. Рассмотрены практические задачи классификации и кластеризации временных рядов мобильных устройств с IMU-датчиков.

Научная новизна.

1. Предложено обобщение метода динамического выравнивания временных рядов на случай множественного выравнивания пространственно-временных объектов.
2. Предложен способ выравнивания непрерывных объектов.
3. Предложен способ взвешенного выравнивания с вычислением значимости измерений временного ряда.
4. Предложен способ выравнивания скрытых представлений.

Теоретическая значимость. Работа в целом носит теоретический характер. Предложен метод множественного выравнивания пространственно-временных объектов. Производится выравнивание при расчете расстояния между объектами более сложной, чем временной ряд, структуры: пространственно-временные объекты. Такой подход определяет функцию расстояния между объектами пространственно-временной структуры и открывает возможность применения метрических методов для поставленной задачи.

Практическая значимость. Предложенные в работе методы предназначены для построения точных, устойчивых и вычислительно простых моделей классификации, кластеризации временных рядов и поиска оптимальной подпоследовательности. Методы предназначены для решения задач распознавания движений человека, поиска паттернов в медицинских временных рядах, классификации видеоряда и других задач. В работе приведены примеры использования предложенных методов для решения задач классификации движений человека по IMU-датчикам и поиска оптимальной подпоследовательности в EEG-сигналах.

Степень достоверности и апробация работы. Достоверность результатов подтверждена математическими доказательствами, экспериментальной проверкой полученных методов на реальных задачах прогнозирования и классификации временных рядов; публикациями результатов исследования в рецензируемых научных изданиях, в том числе рекомендованных ВАК. Результаты работы докладывались и обсуждались на следующих научных конференциях.

1. Динамическое выравнивание непрерывных временных рядов // Ломоносов 2016, 2016.
2. Динамическое выравнивание непрерывных временных рядов // ИОИ-11. 2016.
3. Continuous Time Series Alignment in Human Actions Recognition // AINL FRUCT 2016, 2016.
4. Построение интерпретируемых глубоких моделей в задаче социального ранжирования // Mathematical Methods for Pattern Recognition-18, 2017.
5. Weighted Dynamic Time Warping for optimal subsequence search // EAGE: Тюмень 2019, 2019.

Публикации по теме диссертации. Основные результаты по теме диссертации изложены в 5 печатных изданиях, 5 из которых изданы в журналах, рекомендованных ВАК. Публикации в журналах из списка ВАК.

1. А. В. Гончаров, М. С. Попова, В. В. Стрижов, Метрическая классификация временных рядов с выравниванием относительно центроидов классов // Системы и средства информатики, 2015, том 25, выпуск 4, Сс. 52-64.
2. А. В. Гончаров, В. В. Стрижов, Метрическая классификация временных рядов со взвешенным выравниванием относительно центроидов классов // Информатика и ее применения, 2016, том 10, выпуск 2, Сс. 36-47.
3. A. V. Goncharov and V. V. Strijov, Continuous Time Series Alignment in Human Actions Recognition // Proceedings of the AINL Fruct 2016 Conference, 2016, Pp. 83-86
4. A. V. Goncharov and V. V. Strijov, Analysis of Dissimilarity Set Between Time Series // Computational Mathematics and Modeling, 2018, Vol. 29, Issue 3, Pp. 359-366.
5. А. В. Гончаров, В. В. Стрижов, Выравнивание декартовых произведений упорядоченных множеств // Информатика и ее применения, 2020, том 4, выпуск 1, Сс. 31-39.

Остальные публикации.

6. А. В. Гончаров, В. В. Стрижов, Динамическое выравнивание непрерывных временных рядов // Интеллектуализация Обработки Информации ИОИ-2016, 2016, Сс. 122-123
7. A. V. Goncharov, Weighted Dynamic Time Warping for optimal subsequence search // EAGE, Тюмень 2019, 2019.
8. А. В. Гончаров, Динамическое выравнивание непрерывных временных рядов // Ломоносов-2016, 2016.
9. А. В. Гончаров, В. В. Стрижов, Построение интерпретируемых глубоких моделей в задаче социального ранжирования // Математические методы распознавания образов: Тезисы докладов 18-й Всероссийской конференции с международным участием, 2017, Сс. 20-21.
10. Gleb Morgachev, Alexey Goncharov, Vadim Strijov, Статья с Глебом Distance Function Selection for Multivariate Time-Series // 2019 International Conference on Artificial Intelligence: Applications and Innovations (IC-AIAI), 2019, Pp. 66-664.
11. Evgeny Egorov, Filipp Nikitin, Vasiliy Alekseev, Alexey Goncharov, Konstantin Vorontsov, Topic Modelling for Extracting Behavioral Patterns from Transactions Data // 2019 International Conference on Artificial Intelligence: Applications and Innovations (IC-AIAI), 2020.
12. Vasiliy Alekseev, Evgeny Egorov, Konstantin Vorontsov, Alexey Goncharov, Kaidar Nurumov, Timur Buldybayev, TopicBank: Collection of coherent topics using multiple model training with their further use for topic model validation // Data Knowledge Engineering, 2021, Vol. 135, Pp. 101-921.

13. Tamaz Gadaev, Vladimir Dokholyan, Kirill Tushin, Artsem Salamatsin, Alexey Goncharov, Convolutional Neural Network for Facies Classification and Oil Detection // Saint Petersburg 2020, 2020, Vol. 2020, Is. 1, Pp. 1-5.
14. Victor Bulatov, Vasiliy Alekseev, Konstantin Vorontsov, Darya Polyudova, Eugenia Veselova, Alexey Goncharov, Evgeny Egorov, TopicNet: Making additive regularisation for topic modelling accessible // Proceedings of the 12th Language Resources and Evaluation Conference, 2020, Pp. 6745-6752.

Личный вклад. Результаты получены самостоятельно под научным руководством д.ф.-м.н. В. В. Стрижова. Личный вклад автора в работах с соавторами заключается в следующем: в работах [1–6] сформулированы и получены теоретические результаты.

Объем работы. Диссертация состоит из оглавления, введения, XXX разделов, заключения, списка иллюстраций, списка таблиц, перечня основных обозначений и списка литературы из XXX наименований. Текст работы занимает XXX страниц.

Список сокращений и условных обозначений

| | |
|---------------------|--|
| δ | Множество отклонений пути π |
| γ | Матрица |
| π | Путь между двумя или несколькими пространственно-временными объектами |
| $\rho_\pi(\delta)$ | Функция вычисления расстояния по множеству отклонений |
| θ | Оптимизируемый параметр |
| $d\pi$ | Поправка к пути π |
| $\hat{\mathbf{W}}$ | Множество векторов (либо матриц) весов центроидов |
| \mathbb{I} | Допустимое множество индексов невязок двух (или нескольких) пространственно-временных объектов |
| \mathbb{R}^n | Множество действительных чисел размерности n |
| \mathbb{Y} | Конечное множество классов |
| \mathcal{I}_s | Множество индексов пространственно-временного объекта s |
| \mathcal{S} | Пространство всех дискретных временных рядов |
| \mathcal{S}_c | Пространство всех непрерывных временных рядов |
| \mathfrak{D} | Множество пространственно-временных рядов |
| \mathfrak{R} | Множество функций расстояния |
| \mathfrak{X}_ρ | Признаковое описание объекта, полученное с помощью фиксированной функции расстояния ρ |
| Ω | Тензор отклонений между значениями пространственно-временных объектов |
| $\phi(s_1, s_2, l)$ | Функция вычисления отклонения для элемента пути l между временными рядами s_1 и s_2 |
| C | Множество центроидов |
| c_e | Центроид множества векторов \mathfrak{D}_e |
| l | Элемент пути π |
| S | Пространственно-временной объект, представимый в виде тензора |
| s | Пространственно-временной объект, представимый в виде вектора |
| t | Ось времени |
| x_i | Признаковое описание, присущее объекту |
| $G_w(s)$ | Допустимое преобразование (некоторая последовательность допустимых элементарных преобразований матрицы s) |
| $L(l)$ | Множество возможных элементов, предшествующих элементу пути l |
| $q_\alpha(\delta)$ | α -квантили множества δ |
| T | Граница временного интервала, на котором рассматриваются ряды |
| y_i | Метки классов |

| | |
|-------|--|
| π | Оптимальный путь, выравнивающий путь (путь с наименьшей стоимостью) |
| Cost | Стоимость пути (сумма элементов матрицы Ω , через которые проходит путь) |
| DBA | Dynamic Barycenter Allocation, метод вычисления центроида |
| DTW | Dynamic Time Warping, алгоритм динамического выравнивания оси времени, используется для вычисления расстояния между временными рядами |
| ECoG | Electrocorticography, электрокортикография |
| EEG | Electroencephalography, электроэнцефалография |
| FN | False Negative, ложноотрицательный результат |
| FP | False Positive, ложноположительный результат |
| KNN | k-Nearest Neighbors algorithm, метод k-ближайших соседей |
| MNIST | Modified National Institute of Standards and Technology database, модифицированная база данных Национального Института Стандартов и Технологий |
| mwDTW | matrix-weighted DTW, матрично-взвешенный алгоритм динамической трансформации временной шкалы |
| vwDTW | vector-weighted DTW, векторно-взвешенный алгоритм динамической трансформации временной шкалы |
| WISDM | Wireless Sensor Data Mining lab, датасет лаборатории майнинга данных с беспроводных датчиков |

Список иллюстраций

| | | |
|--------------|---|----|
| Рисунок 1 | Матрица стоимости оптимального выравнивания | 5 |
| Рисунок 3.1 | Алгоритм вычисления поправки $d\pi$ пути π | 36 |
| Рисунок 3.2 | Алгоритм вычисления расстояния между матрицами. . . | 37 |
| Рисунок 6.1 | Выравнивание модельных данных: а) один класс без шума, б) разные классы без шума, в) один класс с шумом, д) разные классы с шумом. | 56 |
| Рисунок 6.2 | Выравнивание данных MNIST: а) один класс, б) разные классы, в) один класс, д) разные классы. | 57 |
| Рисунок 6.3 | Данные спектров сигнала: а) класс 1, б) спектр класса 1, в) класс 2, д) спектр класса 2, е) класс 3, ф) спектр класса 3. . . | 58 |
| Рисунок 6.4 | Примеры синтетических временных рядов аналитических функций (выборка), результаты построения центроида и вектора весов для синтетических данных | 61 |
| Рисунок 6.5 | Примеры синтетических временных рядов аналитических функций (выборка), результат построения матрицы весов центроида класса $\sin(x)$ | 62 |
| Рисунок 6.6 | Примеры временных рядов измерений акселерометра для разных видов физической активности | 63 |
| Рисунок 6.7 | Примеры временных рядов измерений акселерометра для разных видов физической активности | 64 |
| Рисунок 6.8 | Интерполяция кубическими сплайнами | 65 |
| Рисунок 6.9 | Зависимость стоимости выравнивающего пути и времени вычислений от числа узлов | 66 |
| Рисунок 6.10 | Центроиды для шести типов физической активности человека | 69 |
| Рисунок 6.11 | Множество отклонений для евклидова ядра. а) упорядоченное во времени множество. б) Гистограмма распределения. . . | 70 |
| Рисунок 6.12 | Множество отклонений для гауссова ядра. а) упорядоченное во времени множество. б) Гистограмма распределения. . . | 70 |
| Рисунок 6.13 | Матрица попарных расстояний между точками для распознавания жестов. | 74 |
| Рисунок 6.14 | Примеры объектов из датасета используемых электроприборов. | 75 |
| Рисунок 6.15 | Выравнивание попарно пяти временных рядов: а) 1-1, б) 1-2, в) 1-3, д) 1-4, е) 1-5. | 78 |
| Рисунок 6.16 | Выравнивание попарно пяти временных рядов (продолжение): ф) 2-2, г) 2-3, х) 2-4, и) 2-5. | 79 |
| Рисунок 6.17 | Выравнивание попарно пяти временных рядов (продолжение): ж) 3-3, к) 3-4, л) 3-5. | 79 |

| | |
|---|----|
| Рисунок 6.18 Выравнивание попарно пяти временных рядов (продолжение): m) 4-4, n) 4-5. | 79 |
| Рисунок 6.19 Матрица D . | 80 |
| Рисунок 6.20 Матрица A . | 81 |
| Рисунок 6.21 Матрица Z . | 82 |
| Рисунок 6.22 Clustering of 3-dimensional time-series: a) with L_1 based DTW, b) with L_2 based DTW. | 84 |

Список таблиц

| | |
|---|----|
| Таблица 6.1 Снижение расстояний при выполнении преобразований для различных наборов данных. | 59 |
| Таблица 6.2 Сравнение эффективности предложенной (vwDTW) и базовой модели классификации на данных | 63 |
| Таблица 6.3 Сравнение эффективности предложенной (mwDTW) и базовой модели классификации и алгоритма разделяющей классификации на данных | 64 |
| Таблица 6.4 Средние значения для всех классов | 66 |
| Таблица 6.5 Качество многоклассовой классификации временных рядов методами Random forest, SVM, KNN | 71 |
| Таблица 6.6 Результаты классификации временных рядов для задач классификаций активности человека, жестов и электроприборов | 73 |
| Таблица 6.7 Результаты классификации в пространстве представлений для задач классификаций активности человека, жестов и электроприборов | 73 |
| Таблица 6.8 Subsequence Search. | 85 |
| Таблица 6.9 Clustering. | 86 |

Глава 1

Обзор литературы

Временные ряды представляют собой набор измерений, упорядоченных по оси времени. Анализ временных рядов производится при решении задач, связанных с анализом медицинских изображений, классификацией активности человека по измерениям акселерометра телефона, распознаванием речи, анализом видео, поиском паттернов в EEG сигналах (электроэнцефалограмма), кластеризацией набора ECoG (электрокортикограмма) данных и во многих других задачах [7].

Перечислим ряд прикладных задач. Это анализ маммограмм [8], проверка по походке человека наличия у него болезни Альцгеймера с помощью приборов, закрепляющихся на ноге человека и показывающих силу давления стопы на обувь [9], распознавание жестов [10; 11], задача выявления агрессивного поведения водителя на дороге и дорожных аномалий с помощью акселерометра и гироскопа телефона [12], задача аутентификации пользователя по подписи, которую он рисует в воздухе с помощью смартфона с акселерометром [13], задача определения движений человека по данным Microsoft Kinect sensor [14], распознавание произношений цифр в человеческой речи [15], задача определения эмоциональной окраски слов человека [16], выделение слов из речи человека без учителя [17], задача выявления различных аритмий у пациента по его кардиограмме [18], выравнивание видео с его текстовым описанием [19], онлайн мониторинг производства [20].

Рассматриваются задачи бинарной и многоклассовой классификации [8; 14], распознавания речи [15–17], выравнивания и сегментации видео [19].

Исследуемые объекты представляют собой одномерные и многомерные временные ряды. В большинстве случаев они предварительно нормализуются и пропускаются через фильтр низких частот. Для ускорения работы используется даунсэмплинг временных рядов [18]. В случае [8], когда исходные объекты представляют собой двумерные изображения, они предварительно растягиваются в один вектор для получения временного ряда.

В [8] в качестве расстояния между последовательностями берется DTW-расстояние с Sakoe-Chiba band. Далее обучаются KNN-классификаторы с различными К (1, 3, 5, 7, 11) и подбором порогового количества соседей (из К ближайших) из одного класса, по которому принимается решение о классе интересующего объекта. Выбор происходит следующим образом: из обучающей выборки берется объект, и находится значение, для которого этот объект будет правильно классифицирован с помощью KNN, так происходит для различных объектов обучающей выборки и для различных обучающих выборок, таким образом обучается значение порогового числа соседей. Все это делается для достижения оптимального соотношения между FP и FN.

В [9] изначально походка человека представлялась в виде временного ряда. Далее походка каждого человека представлялась в виде двумерного при-

знакового вектора, компоненты которого вычислялись следующим образом. По данным с приборов с помощью функции midcross определялись времена шагов. Выяснилось, что у пациентов, больных болезнью Альцгеймера разница между временами шагов в среднем выше, чем у здоровых людей, поэтому в качестве первого признака бралась дисперсия времени шага. Также сравнивались начальные и конечные отрезки исходных временных рядов между собой посредством вычисления DTW расстояния между ними, которое бралось в качестве второго признака. Такой выбор признака мотивирован тем, что походка человека с болезнью Альцгеймера более неравномерна. Далее по построенному набору векторов проводилась классификация.

В [10] в каждом жесте разные точки тела человека вносят разный вклад в жест (например при приветствии правой рукой очевидно, что нет смысла учитывать движения левой руки), поэтому для классификации жестов используется собственный вариант взвешенного DTW. Веса для учета различных точек в DTW-расстоянии зависят от суммарного расстояния, пройденного конкретной точкой во время измерений, которое зажимается между двумя пороговыми значениями (threshold). Также веса имеют параметр, который подбирается таким, чтобы вариация внутри классов была минимальна, а между классами максимальна, что достигается с помощью максимизации дискриминантного значения (аналогично линейному дискриминанту Фишера).

В [21] используется метод DTW, выполняемый параллельно на нескольких ядрах. Обучающая выборка распределялась равномерно между несколькими ядрами, на которых впоследствии запускался DTW алгоритм, в конце сравнивались результаты каждого ядра и в качестве ответа брался наилучший.

В [11] для классификации используется стандартный ND-DTW, который распараллеливается для ускорения работы. Также определяется пороговое значение для отсеивания жестов, не принадлежащих ни одному из классов.

В [12] были проведены два эксперимента: сравнение DTW и KNN при классификации с 3 классами: нормальное вождение, агрессивное вождение, дефект дорожного покрытия, и данными (a_x, a_y, a_z) ; а также сравнение DTW и KNN при классификации с 2 классами: нормальное вождение, агрессивное вождение, и данными a_x, g_z .

В [13] были использованы различные подходы: MGHMM (mixtures of Gaussian hidden Markov models), байесовский классификатор и несколько измененный DTW, названный Average DTW. В Average DTW изначально из обучающей выборки одного класса берется образец, который в среднем ближе ко всем остальным, затем с помощью обычного DTW вычисляются выравнивающие пути со всеми остальными образцами и образцы сжимаются (растягиваются) во времени соответствующим образом, при этом вместо двух склеивающихся точек создается одна усредненная. Итоговый временной ряд, представляющий класс, получается усреднением получившихся рядов, далее проводится 1-NN классификация.

В [14] и [18] используется классический 1NN-DTW для определения клас-

са временного ряда: по матрице попарных расстояний осуществляется поиск ближайшего похожего объекта выборки.

В [15] из ряда извлекаются MFCC признаки с помощью фильтрации, деления на фреймы, окна Хэмминга, FFT, Мел-фильтрации, etc, а также их скорости и ускорения. Также описывается новый метод конструирования признаков — WMFCC (weighted MFCC), комбинирующий каждый признак с его скоростью и ускорением с некоторыми весами. Классификация полученных временных рядов проводится с помощью DTW, DDTW, а также IFDTW (improved features DTW), где конструируются новые признаки — локальные и глобальные. IFDTW используется с урезанным на 10% по диагонали Itakura parallelogram, ребра которого имеют коэффициенты наклона 3 и 1/3.

В [16] из сырых временных рядов конструируются MFCC признаки, а также скорость и ускорение и далее применяется SVM (Support Vector Machine, метод опорных векторов) с DTW-расстоянием.

В [17] исходный временной ряд делится на фреймы и строится матрица DTW-расстояний между фреймами (представляемыми в виде PLP (Parametric Linear Programming) признаков с 1-ми и 2-ми производными), выделяются множества DTW-схожих элементов матрицы (слова), которые обычно представляют собой диагональные отрезки матрицы, далее они кластеризуются. Для каждого кластера его элементы выравниваются между собой с помощью DTW, где расстояния между фреймами считаются с помощью коэффициента Оти-аи, элементы же разных кластеров также выравниваются между собой, но по диагонали (для разноса кластеров между собой). Полученные слова, подаются на вход сиамской нейронной сети, для которой определяется такая функция потерь, чтобы в ходе обучения сеть максимизировала расстояние между словами разных классов и минимизировала его между словами одинаковых классов. Сеть выдает на выходе два 100-мерных вектора, расстояние между которыми считается с помощью симметричной дивергенции Кульбака-Лейблера.

В [19] для решения поставленных задач предлагается дискриминативная модель, использующая принцип DTW в том, что видеоряд выравнивается относительно опорного скрипта по соответствующей матрице расстояний по пути минимальной стоимости. Вместо функции минимизации используется её сглаженная версия, позволяющая получить дифференцируемую функцию потерь. Функция расстояния в DTW обучается с помощью RNN (Recurrent Neural Network).

В [20] идея на примере одного временного ряда состоит в следующем: системе мониторинга изначально доступен временной ряд, полученный при изготовлении золотой выборки, ей на вход в онлайн-режиме поступает временной ряд текущего изделия. Создается матрица DTW расстояний между всевозможными префиксами имеющихся временных рядов, по ней строится временной ряд ошибки, каждый элемент которого вычисляется как минимум по соответствующей строке матрицы с учетом warping constraint. Этот временной ряд представляет собой аккумулированную ошибку и используется для локализа-

ции ошибки. Если он превышает заранее определенную константу, то система сигнализирует об отклонении от нормы.

В настоящей работе рассматриваются метрические методы классификации временных рядов. Метрические методы используют информацию о расстоянии между объектами в наборе данных. Качество метрических методов зависит от выбора функции расстояния между объектами.

В литературе используются различные функции расстояния: расстояние Хаусдорфа [22], MODH [23], расстояние, основанное на HMM (hidden Markov model) [24], Евклидово расстояние в исходном пространстве или в пространстве сниженной размерности [25], метод динамического выравнивания временных рядов [26; 27], LCSS (Longest Common SubSequence) [28; 29], Edit Distance with Real Penalty [30], Edit Distance on Real sequence [31], DISSIM (Dissimilarity metric, метрика отклонений) [32], Sequence Weighted Alignment model [33], Spatial Assembling Distance [34] и другие.

В качестве базового метода для построения функции расстояния в настоящей работе предлагается использовать динамическое выравнивание временных рядов (Dynamic Time Warping) [26; 35]. Как показано в [36], этот метод находит наилучшее соответствие между двумя временными рядами, если они нелинейно деформированы друг относительно друга — растянуты, сжаты или смещены вдоль оси времени. Следовательно, функция расстояния, основанная на алгоритме динамического выравнивания временных рядов DTW, более устойчива к локальным или глобальным деформациям оси времени [37; 38] и дает лучшее качество классификации по сравнению с другими функциями расстояния [32–34; 39].

На данный момент метод DTW является наиболее часто используемым методом для расчета расстояния между двумя временными рядами и имеет множество модификаций в литературе.

DTW с использованием ограничений Сакое-Чиба или Итакура [40] (Sakoe-Chiba band) позволяет задать ограничения на выравнивающий путь (определяющий соответствие между двумя временными рядами) и ускорить вычисление расстояния. При этом в литературе существует большое число верхних и нижних оценок [41] на вычислительную сложность модификаций алгоритма DTW.

Использование ядер (Kernel-DTW) [42] улучшает качество работы метода DTW. Эта модификация метода DTW использует ядра, такие как Гауссовское [43; 44], для того чтобы вычислять расстояние между парой значений двух временных рядов. В недавних статьях [42; 44; 45] предлагается вычислять расстояние между временными рядами как взвешенную сумму значений всех возможных выравнивающих путей, полученных с использованием ядерных функций, вместо использования одного выравнивающего пути между двумя объектами. Показано, что такой подход позволяет получить более высокое качество решение исходной задачи, так как является более устойчивым к локальным и глобальным изменениям в объектам.

Поиск всех попарных расстояний между объектами в задачах метрического

анализа данных является вычислительно трудоемкой задачей. Для снижения объема вычислений и снижения размерности получаемой матрицы попарных расстояний в настоящей работе предлагается использовать подход с применением расстояний между объектами выборки и эталонными объектами классов, кластеров или всей выборки. В таком подходе матрица попарных расстояний имеет меньшую размерность, так как расстояние вычисляется между объектами выборки и небольшим количеством эталонных объектов.

Предпосылкой для такого подхода является гипотеза о природе данных, которую мы принимаем при решении задач классификации или кластеризации: предполагается схожесть объектов одного класса или кластера в исходном признаковом пространстве. Уточним это предположение: в одном и том же классе находятся временные ряды, имеющие схожую форму с точностью до линейной или нелинейной деформации, локальных или глобальных сдвигов по оси времени. В таком случае в качестве эталонного объекта может быть использован один объект, и он сможет заменить использование всех объектов одного класса для вычисления матрицы попарных расстояний.

Базовым методом решения задач в работе принимается метрический подход с последующим решением задач классификации, кластеризации и поиска подпоследовательности на основе матрицы попарных расстояний между объектами. Данный метод описан в работе [1]. В работе эталонным объектом класса выбирается объект выборки, который является ближайшим ко всем остальным объектам выбранного класса.

Существует альтернативный способ выбора такого эталонного объекта - это метод вычисления центроида, который позволяет вычислить его точно. Это метод DBA, решающий задачу оптимизации для нахождения центроида. Алгоритм применения и доказательство корректности приведены в [46].

После того как центроиды классов построены, классификация производится в соответствии с методом ближайшего эталонного объекта (или центроида). Таким образом, базовая модель классификации состоит из трех шагов. Первый — отбор эталонных объектов каждого класса. Второй — построение матрицы попарных расстояний сниженной размерности между временными рядами и эталонными объектами каждого класса. Третий — классификация временных рядов методом ближайшего центроида с помощью матрицы попарных расстояний.

В работах автора [2–6] предлагаются различные обобщения метода DTW.

В работе [3] предлагается перейти от рассмотрения объекта, временного ряда, к более общему случаю, в котором объект имеет несколько осей времени. Предлагается рассмотреть частный случай — объекты, определенные на двух осях времени, который считаются независимыми. В случае единственной дискретной и ограниченной сверху шкалы времени объект представим в виде матрицы. Таким образом, частным случаем метода, исследованного в работе [3], является метод выравнивания двух матриц друг относительно друга.

Вводятся ограничения на зависимости осей времени в декартовом произве-

дении для таких объектов. Определена гипотеза порождения данных: объекты одного класса эквивалентности получены при помощи допустимых преобразований, а именно локальных деформаций — растяжений и сжатий каждой из осей времени по отдельности. В дискретном случае преобразование представимо дублированием строк и столбцов матриц. В число допустимых преобразований попадают и глобальные деформации: сдвиги по осям времени, представимые добавлением и удалением крайних строк и столбцов исходных матриц. Для каждой из осей времени выполняются свойства времени: монотонность и непрерывность. Похожими на описанные свойствами обладает объект, например, частотный спектр сигнала, где одна ось определяет время, а другая — частоту, величину, обратную времени.

Между двумя объектами, матрицами, в случае допустимых преобразований требуется определить инвариантную к преобразованиям осей времени функцию расстояния, которая сможет выделить классы эквивалентности множества преобразованных объектов. Работа [3] посвящена определению такой функции расстояния как обобщение метода динамического выравнивания временных рядов DTW для матриц.

Классический метод динамического выравнивания временных рядов определен только для объектов с одной осью времени, что делает его неприменимым для описанного случая. Однако концепции, используемые на каждой стадии вычисления оптимального выравнивания, обобщены на рассматриваемый случай. Работа исследует свойства предложенного метода и сравнивает результаты применения метода к задачам классификации изображений [47] с результатами функции расстояния L_2 .

Для иллюстрации и анализа результатов решается задача метрической классификации объектов (матриц низкой размерности). Используются наборы данных: модельные данные, которые согласуются с выдвинутой гипотезой порождения данных для временных рядов, подмножество датасета MNIST сниженной размерности и частотный спектр сигнала.

В [2] вводятся понятия вектора весов и матрицы весов центроида и описываются методы vwDTW и mwDTW (векторно- и матрично-взвешенный DTW) вычисления функции расстояния. Предполагается, что в центроиде присутствуют характерные для всего класса участки, которым соответствуют большие веса вектора весов этого центроида. Поэтому функция расстояния, основанная на vwDTW и использующая вектор весов центроида, точнее объединит объекты одного класса и разделит объекты разных классов, чем основанная на DTW. Поэтому в предлагаемой модели используется метод vwDTW как для вычисления центроида по методу DBA, так и для построения матрицы попарных расстояний. Также исследуются свойства, вид матрицы весов центроида и эффективность применения расстояния, вычисленного с ее помощью — mwDTW, в прикладных задачах.

Для проверки работоспособности такой модели в [2] проведен вычислительный эксперимент на реальных и синтетических данных. Эксперимент включает

в себя анализ и классификацию данных при помощи построенной модели. Полученные результаты сравниваются с результатами применения базовой модели к тем же исходным данным.

Представление дискретных временных рядов с помощью непрерывных объектов — полезный подход для анализа временных рядов, имеющих различные масштабы. Многомасштабные временные ряды часто встречаются на практике, например, в медицине, индустрии или даже в финансовых приложениях: информация о состоянии здоровья человека [48; 49], сигналы ЭЭГ [50], данные движения человека [51] и финансовые данные [52] собираются на различных частотах, которые могут меняться в десятки раз [53]. Следовательно, такие измерения сложно анализировать, используя только их дискретные представления.

В работе [4] предлагается метрический метод анализа непрерывных временных рядов, основанный на метрике DTW. Этот подход позволяет уменьшить ограничения на количество используемой памяти с помощью представления дискретных объектов их моделями и на время вычислений — с помощью аналитических оценок.

Алгоритм динамического выравнивания DTW определяет расстояние между дискретными временными рядами. Динамическое программирование помогает найти выравнивающий путь в дискретном случае [35]. Невозможно использовать стандартный метод DTW для непрерывных объектов. Мы вводим понятия DTW расстояния между непрерывными временными рядами, выравнивающего пути между ними и его стоимости. Мы решаем задачу поиска оптимального пути, приближая настоящий путь параметрическими функциями. В-сплайны или кубические сплайны [54], например, могут быть применены для этой аппроксимации. Задача поиска выравнивающего пути сводится к задаче поиска наилучшей аппроксимации или, другими словами, наиболее подходящих параметров. Универсальность этого подхода позволяет использовать любой тип аппроксимации выравнивающего пути.

В [4] предлагается новый подход к анализу непрерывных временных рядов. Свойства выравнивающего пути и его стоимости исследуются в экспериментальной части работы. Эта часть использует дискретные временные ряды, полученные с измерений акселерометра мобильного телефона [55]. Их параметрическая аппроксимация используется для получения непрерывных объектов. Мы используем приближение кубическими сплайнами. Для того чтобы описать способность новой функции расстояния в пространстве непрерывных функций разделять объекты различных классов, решается задача метрической классификации. Строятся центроиды классов и классификация производится согласно методу ближайшего центроида.

В [6] функция расстояния между временными рядами используется для их признакового описания. Размерность полученного с помощью метрики признакового пространства должна быть намного меньше размерности исходных объектов.

Вектор-функция расстояния между данным временным рядом и некоторым набором других объектов дает признаковое описание временного ряда [35]. Ее вычисление для всех пар временных рядов из выборки — вычислительно сложная задача. Она вычисляется только между временным рядом и эталонными объектами: центроидами классов [46] или центрами кластеров.

В работе [6] функция расстояния, используемая для построения признакового пространства, — это отображение из пространства пар объектов в пространство вещественных чисел с размерностью, отличной от единицы. Это позволяет улучшить качество классификации. Предложенный метод применим к модифицированным версиям алгоритма динамического выравнивания временных рядов DTW [36], где анализируются отклонения между ними.

Вычислительный эксперимент проводится на датасете WISDM [55]. Временные ряды были нормализованы и центроиды были построены для каждого класса. Алгоритм составляет выравнивающий путь для каждой пары временной ряд — центроид в соответствии с выбранными ограничениями и фиксированным ядром. Он составляет множество отклонений (dissimilarities set), используя выравнивающий путь. Среднее по множеству отклонений соответствует расстоянию DTW, а квантили распределения элементов этого множества и средние значения распределения на задаваемых квантилями интервалах соответствуют предлагаемой вектор-функции. Методы классификации используют полученную матрицу признаков. Качество классификации при использовании дополнительных статистик улучшилось по сравнению с качеством классификации с использованием только среднего значения.

Глава 2

Обобщение множественного выравнивания и построение центроида

Цель данной работы — составить единое представление теории динамического выравнивания пространственно-временных объектов. Подобные объекты могут иметь разную природу, однако единое представления позволяет работать с ними, используя одну математическую базу и обоснованность используемых методов.

2.1. Множественное выравнивание временных рядов

Для начала введем единые определения и обозначения, которые будут использоваться в работе.

Начнем с выборки $\mathfrak{D} = \{\mathbf{S}_i(\mathbf{t})\}_{i=1}^N$ — множество пространственно-временных рядов. \mathbf{S}_i обозначает пространственно-временной объект с индексом i и в зависимости от конкретного случая может лежать в пространствах различной размерности и быть представленным как тензором \mathbf{S} , так и вектором \mathbf{s} ; \mathbf{t} обозначает ось времени и в общем случае также является многомерным вектором.

Приведем некоторые примеры объектов и их обозначений:

Определение 1. Будем называть временным рядом последовательность действительных чисел $\mathbf{s}(t) = (s_1, \dots, s_n)$. Временной ряд также представим в виде вектора $\mathbf{s}(t) \in \mathbb{R}^n$.

Определение 2. Временной ряд на двух осях времени — это последовательность $\mathbf{s}(t_1, t_2) = (s_{1,1}, \dots, s_{1,n}, \dots, s_{n,1}, \dots, s_{n,n})$. Он также представим в виде матрицы $\mathbf{s}(t_1, t_2) \in \mathbb{R}^{n \times n}$.

Определение 3. Многомерный временной ряд $\mathbf{S}(t)$ — это последовательность $(\mathbf{s}_1, \dots, \mathbf{s}_n)$, где $\mathbf{s}_i \in \mathbb{R}^m$ — вектор. Он представим матрицей $\mathbf{S}(t) \in \mathbb{R}^{n \times m}$.

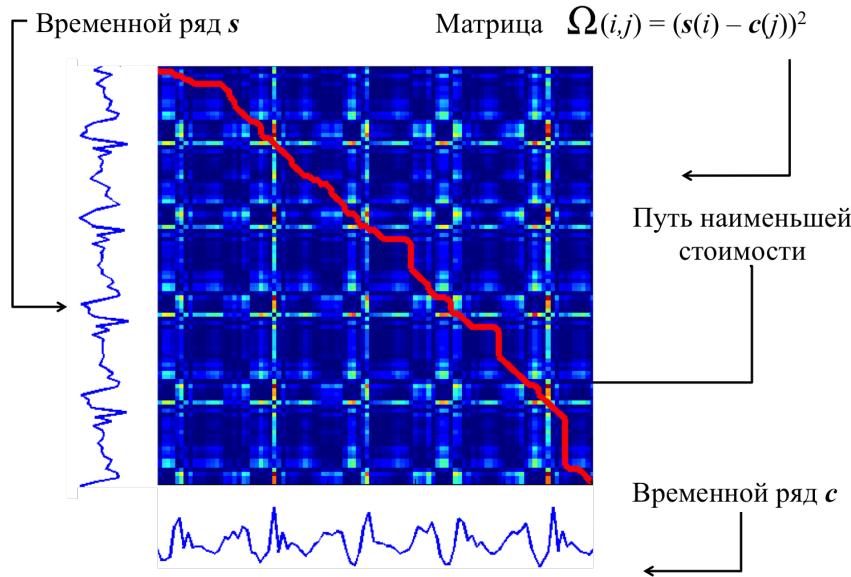
Определение 4. Будем обозначать \mathcal{I}_s — множество индексов пространственно-временного объекта \mathbf{s} .

Для приведенных выше примеров объектов множество индексов будет представлено как:

- Для временного ряда $\mathbf{s}(t)$: $\mathcal{I}_{\mathbf{s}(t)} = \{1, \dots, n\}$;
- Для временного ряда на двух осях времени $\mathbf{s}(t_1, t_2)$: $\mathcal{I}_{\mathbf{s}(t_1, t_2)} = \{(1, 1), \dots, (1, n), \dots, (n, 1), \dots, (n, n)\}$;
- Для многомерного временного ряда $\mathbf{S}(t)$: $\mathcal{I}_{\mathbf{S}(t)} = \{1, \dots, n\}$.

В работе мы используем метрические методы, которые работают с двумя объектами из пространства. Поэтому важным понятием для исследования является множество невязок $\boldsymbol{\delta}$, о котором речь пойдет ниже. Чтобы его определить необходимо дать определение невязки.

Определение 5. Невязка — это количественно описанное различие между элементами различных пространственно-временных объектов. Таким образом, $\mathcal{I}_{\mathbf{s}_1} \times \mathcal{I}_{\mathbf{s}_2}$ — индексы всех невязок двух пространственно-временных объектов \mathbf{s}_1 и \mathbf{s}_2 .



Аналогично, для пространств, объекты которых были приведены в качестве примеров выше, индексы всех невязок будут выглядеть как:

$$\begin{aligned}\mathcal{I}_{\mathbf{s}(t)} \times \mathcal{I}_{\mathbf{s}(t)} &= \{(1, 1), \dots, (1, n), \dots, (n, 1), \dots, (n, n)\}; \\ \mathcal{I}_{\mathbf{s}(t_1, t_2)} \times \mathcal{I}_{\mathbf{s}(t_1, t_2)} &= \{(1, 1, 1, 1), \dots, (1, 1, 1, n), \dots, (n, 1, 1, 1), \dots, (n, n, n, n)\}; \\ \mathcal{I}_{\mathbf{S}(t)} \times \mathcal{I}_{\mathbf{S}(t)} &= \{(1, 1), \dots, (1, n), \dots, (n, 1), \dots, (n, n)\}.\end{aligned}$$

Невязки могут быть определены даже не на двух, а на нескольких объектах. В этом случае индексы всех невязок для трех объектов будет выглядеть как:

$$\mathcal{I}_{\mathbf{s}(t)} \times \mathcal{I}_{\mathbf{s}(t)} \times \mathcal{I}_{\mathbf{s}(t)} = \{(1, 1, 1), \dots, (1, 1, n), \dots, (n, 1, 1), \dots, (n, n, n)\}.$$

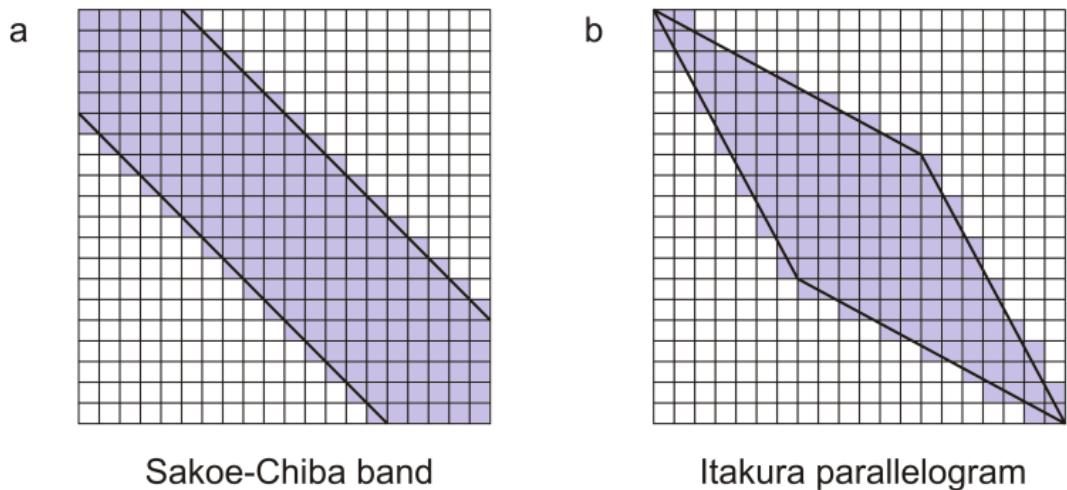
Для выравнивания двух объектов друг относительно друга необходимо проложить путь по множеству невязок. Но возможно искусственно ввести ограничения на множество тех невязок, по которым может пройти этот самый путь.

Определение 6. Обозначим $\mathbb{I} \subseteq \mathcal{I}_{\mathbf{s}_1} \times \mathcal{I}_{\mathbf{s}_2}$ — допустимое множество индексов невязок двух временных рядов \mathbf{s}_1 и \mathbf{s}_2 (которое является **гиперпараметром алгоритма**).

Определение 7. $\pi = (\pi_1, \pi_2, \dots, \pi_K) \subset \mathbb{I}$ — путь между пространственно-временными объектами, свойства которого зависят от конкретного случая (на путь накладываются определенные свойства, согласно которым он может быть определен однозначно внутри допустимого множества индексов невязок). Путь представим в виде подмножества индексов. Допустимого множества индексов невязок, которое обладает определенными свойствами и должно соответствовать ограничениям. Здесь K — длина пути. Допустимый путь должен удовлетворять следующим условиям:

Границные условия: $\pi_1 = (1, \dots, 1)$, $\pi_K = (n, \dots, n)$.

Непрерывность: $\pi_k = (i, j, \dots)$, $\pi_{k-1} = (i', j'), \dots$, тогда $|i - i'| \leq 1$, $|j - j'| \leq 1$. Это условие означает, что соседние вершины пути находятся рядом.



Монотонность: $\pi_k = (i, j, \dots)$, $\pi_{k-1} = (i', j', \dots)$, тогда $i - i' \geq 0$, $j - j' \geq 0$. Это условие гарантирует, что путь не вернется в предыдущие вершины.

Определение 8. Обозначим $\mathbf{l} = \boldsymbol{\pi}_i \in \boldsymbol{\pi} = \{\boldsymbol{\pi}_i\}_{i=1}^{|\boldsymbol{\pi}|}$ — как один элемент пути $\boldsymbol{\pi}$.

Для последующего вычисления расстояния между объектами будут использоваться методы динамического программирования, где активно используется определение:

Определение 9. $L(\mathbf{l}) \subseteq \mathbb{I}$ — множество возможных элементов, предшествующих элементу пути \mathbf{l} . Это множество влияет на вычисление пути между пространственно-временными объектами и является **гиперпараметром алгоритма**.

Для пространств, объекты которых были приведены в качестве примеров выше, примеры множества возможных элементов, предшествующих элементу пути, будут выглядеть как:

$$\begin{aligned} L(i, j) &= \{(i-1, j), (i, j-1), (i-1, j-1)\}; \\ L(i, j, k, l) &= \{(i-1, j, k, l), (i, j-1, k, l), (i, j, k-1, l), (i, j, k, l-1), \\ &\quad (i-1, j, k-1, l), (i, j-1, k, l-1), (i-1, j-1, k-1, l-1)\}. \end{aligned}$$

Количественное описание различий между элементами двух (или нескольких) пространственно-временных объектов, или невязка, рассчитывается на основе значений этих самых элементов. Вычисление точного значения невязки производится при помощи функции:

Определение 10. Функция отклонения $\phi(\mathbf{s}_1, \mathbf{s}_2, \mathbf{l})$ для элемента пути \mathbf{l} между временными рядами \mathbf{s}_1 и \mathbf{s}_2 влияет на путь между временными рядами и является **гиперпараметром алгоритма**.

Приведем примеры функций вычисления отклонений, которые активно используются в литературе:

$$\phi(\mathbf{s}_{1l_1}, \mathbf{s}_{2l_2}) = (\mathbf{s}_{1l_1} - \mathbf{s}_{2l_2})^2; \quad (2.1)$$

$$\phi(\mathbf{s}_{1l_1}, \mathbf{s}_{2l_2}) = e^{\frac{1}{2\pi}(\mathbf{s}_{1l_1} - \mathbf{s}_{2l_2})^2}; \quad (2.2)$$

$$\phi'(\mathbf{s}_{1l_1}, \mathbf{s}_{2l_2}) = \mathbf{k}(l_1, l_2) \phi(\mathbf{s}_{1l_1}, \mathbf{s}_{2l_2});$$

где \mathbf{k} — дополнительный параметр алгоритма, который соответствует вкладу невязки в общее расстояние между объектами.

С определением этой функции вычисления отклонений мы можем более точно определить:

Определение 11. Множество отклонений $\boldsymbol{\delta}_\pi = \{\phi(\mathbf{s}_1, \mathbf{s}_2, \boldsymbol{\pi}_i)\}_{i=1}^{\|\boldsymbol{\pi}\|}$ пути $\boldsymbol{\pi}$.

При этом мы можем определить и следующее понятие:

Определение 12. Тензор отклонений $\Omega_l = \phi(\mathbf{s}_1, \mathbf{s}_2, \mathbf{l})$ между значениями пространственно-временных объектов.

Наличие множества отклонений между пространственно-временными объектами позволяет ввести функцию расстояния между этими объектами:

Определение 13. Функция вычисления расстояния $\rho(\boldsymbol{\delta}_\pi)$ по множеству отклонений или же стоимости пути (функция вычисления расстояния влияет на вычисление расстояния между временными рядами и является гиперпараметром алгоритма):

$$\rho : \mathbb{R}^{\|\boldsymbol{\pi}\|} \rightarrow \mathbb{R}^l, \quad (2.3)$$

где l — размерность пространства, в которое происходит отображение вектор-функции расчета расстояния.

Требуется определить расстояние между двумя объектами. Основа метода состоит в том, что за расстояние между двумя объектами берется наименьшее значение расстояния, которое можно получить по какому-либо из путей между объектами. Оно соответствует оптимальному пути:

Определение 14. Оптимальный путь $\hat{\boldsymbol{\pi}}$ — это путь, расстояние по множеству невязок которого $\rho(\boldsymbol{\delta}_{\hat{\boldsymbol{\pi}}})$ — наименьшее из всех возможных.

$$\hat{\boldsymbol{\pi}} = \underset{\pi}{\operatorname{argmin}} \rho(\boldsymbol{\delta}_\pi).$$

За расстояние DTW между пространственно-временными объектами принимается расстояние, вычисленное по множеству невязок оптимального пути. Для поиска такого используется:

Определение 15. Алгоритм выбора оптимального (или выравнивающего) пути между пространственно-временными объектами:

$$\begin{aligned} \mathbf{a} : (\mathbf{s}_1, \mathbf{s}_2) &\mapsto \hat{\boldsymbol{\pi}}, \\ \mathbf{a} : \mathbb{R}^n \times \mathbb{R}^n &\rightarrow \mathbb{I}^{\|\hat{\boldsymbol{\pi}}\|}. \end{aligned} \quad (2.4)$$

2.1.1. Базовый алгоритм вычисления расстояния методом DTW

Изложим базовый алгоритм динамического выравнивания, который определяет порядок вычисления расстояния по методу DTW для двух одномерных временных рядов одинаковой размерности.

Пусть $\mathbf{s}_1, \mathbf{s}_2$ — два одномерных временных ряда длины n . Зафиксируем $\mathbb{I} = \mathcal{I}_{\mathbf{s}_1} \times \mathcal{I}_{\mathbf{s}_2} = \{(1, 1), \dots, (1, n), \dots, (n, 1), \dots, (n, n)\}$.

Стоимостью пути примем сумму элементов матрицы невязок Ω , через которую проходит путь:

$$\text{Cost}(\mathbf{s}_1, \mathbf{s}_2, \pi) = \sum_{(i,j) \in \pi} \Omega_{ij}.$$

Алгоритм выбора оптимального пути состоит в том, чтобы выбрать путь с наименьшей стоимостью.

Существует рекуррентный алгоритм поиска оптимального пути. Определим матрицу γ той же размерности, что и матрица Ω :

$$\begin{aligned} \gamma_{1j} &= \Omega_{1j}, \quad \gamma_{i1} = \Omega_{i1}, \\ \gamma(i, j) &= \Omega_{ij} + \min\{\gamma(i-1, j-1), \gamma(i-1, j), \gamma(i, j-1)\}. \end{aligned}$$

Несложно видеть, что элемент γ_{ij} матрицы γ равен стоимости оптимального выравнивающего пути между первыми i значениями s_1 и первыми j значениями s_2 .

Стоимостью пути является элемент γ_{nn} , который можно получить, проведя все требуемые вычисления для элементов матрицы. Именно стоимость пути примем за значение расстояния DTW между объектами \mathbf{s}_1 и \mathbf{s}_2 .

СЮДА ВСТАВИТЬ АЛГОРИТМ

2.1.2. Общий алгоритм вычисления расстояния методом DTW

Сформулируем теперь общий алгоритм поиска расстояния DTW между двумя пространственно-временными объектами \mathbf{S}_1 и \mathbf{S}_2 . Индексами всех возможных невязок в таком случае является $\mathcal{I}_{\mathbf{S}_1} \times \mathcal{I}_{\mathbf{S}_2}$.

Для алгоритма необходимо определить допустимое множество индексов невязок, внутри которого и будет происходить поиск оптимального пути: $\mathbb{I} \subseteq \mathcal{I}_{\mathbf{S}_1} \times \mathcal{I}_{\mathbf{S}_2}$

Также необходимо определить и функцию ϕ . С ее помощью получится вычислить тензор отклонений Ω следующим образом: для всех $\mathbf{l} \in \mathbb{I} \subseteq \mathcal{I}_{\mathbf{S}_1} \times \mathcal{I}_{\mathbf{S}_2}$: $\Omega_{\mathbf{l}} = \phi(\mathbf{S}_1, \mathbf{S}_2, \mathbf{l})$.

Определим новый тензор Γ , размерность которого соответствует тензору Ω . Каждый элемент тензора Γ будет содержать множество отклонений оптимального выравнивающего пути от первого до соответствующего элемента тензора ω . Для вычисления значений нового тензора используются следующие формулы:

$$\Gamma_{1, \dots, 1, j} = \{1, \dots, 1, j\}, \dots, \Gamma_{i, 1, \dots, 1} = \{i, 1, \dots, 1\};$$

Для всех $\mathbf{l} \in \mathbb{I}$

$$\Gamma(\mathbf{l}) = \mathbf{l} \cup \arg \min_{\mathbf{i} \in L(\mathbf{l})} \rho(\cup_{\mathbf{j} \in \Gamma_{\mathbf{i}}} \Omega_{\mathbf{j}}).$$

Крайний элемент тензора Γ содержит выравнивающий путь между двумя пространственно-временными объектами:

$$\hat{\pi} = \Gamma_{n,\dots,n}.$$

Определим расстояние DTW как функцию от множества отклонений этого пути: $\rho(\delta_{\Gamma_{n,\dots,n}})$.

Весь описанный выше алгоритм **a** выбора выравнивающего пути как и ранее заключается в том, чтобы выбирать путь с наименьшей стоимостью для последующего вычисления расстояния по множеству невязок. Алгоритм можно определить как $a_{\mathbb{I},L,\phi,\rho}(S_1, S_2)$.

СЮДА ВСТАВИТЬ АЛГОРИТМ

2.2. Постановка задачи выбора функции расстояния с точки зрения классов эквивалентности.

Заданы объекты вида $S(t) \in \mathbb{R}^{n \times m}$. Функция $G_w(s) : \mathbb{R}^{n \times m} \rightarrow \mathbb{R}^{\hat{n} \times \hat{m}}$ задает допустимые параметрические преобразования исходного объекта s : глобальные сдвиги, локальные линейные и нелинейные деформации, а именно растяжения и сжатия оси времени, сдвиги значений по оси времени. Скалярный параметр $w \in \mathbb{R}^+$ функции G фиксирует набор этих преобразований.

Допустимым элементарным преобразованием объекта S назовем операции с элементами $\{S_i\} \in S$, например: дублирование элемента, удаление элементов и добавление случайного шума к элементу. Допустимым преобразованием примем некоторую последовательность допустимых элементарных преобразований объекта S и обозначим как $G_w(S)$. **ДОПИСАТЬ ФОРМАЛЬНЫЕ ОПРЕДЕЛЕНИЯ ОПЕРАЦИЙ С ЭЛЕМЕНТАМИ**

Будем называть объект $S' \in \mathbb{R}^{\hat{n} \times \hat{m}}$, полученным из объекта S при помощи допустимых преобразований $G_{\hat{w}}$, если существует $\hat{w} \in \mathbb{R}^+ : S' = G_{\hat{w}}(S)$.

Функцию расстояния между двумя объектами $\rho : \mathbb{R}^{n \times m} \times \mathbb{R}^{\hat{n} \times \hat{m}} \rightarrow \mathbb{R}^+$ определим на выборке $\mathfrak{D} = \{S_i\}_{i=1}^m$ объектов вида $S_i \in \mathbb{R}^{n \times m}$.

Определение 16. Под классом эквивалентности $\{S'_j\}$ объекта $S'_i \in \mathfrak{D}$ будем понимать все объекты $S' \in \mathfrak{D} | \exists w_i, w_j : G_{w_i}(S_i) = G_{w_j}(S'_j)$. То есть существуют допустимые трансформации с параметрами w_i и w_j , такие, что $G_{w_i}(S_i) = G_{w_j}(S'_j)$.

Стоит заметить, что каждый объект выборки может принадлежать только одному классу эквивалентности: для объекта S_i и объекта S''_k из других классов эквивалентности $\{S''_k\}_i = \{s'' \in \mathfrak{D} | \nexists w_i, w_k : G_{w_i}(S_i) = G_{w_k}(S'')\}$, не существует таких w_i, w_k , что выполняется $G_{w_i}(S_i) = G_{w_k}(S'')$.

Определение 17. Решается задача поиска функции расстояния ρ , значение которой на паре объектов одного класса эквивалентности меньше, чем на любой паре объектов из разных: для любого $i, j, k \in \{1, \dots, m\}$ $\rho(s_i, s'_j) < \rho(s_i, s''_k)$. Функцию расстояния, обладающую таким свойством назовем инвариантной на классах эквивалентности.

Возможно определить критерий качества функции расстояния ρ на выборке \mathfrak{D} как доля объектов, для которых выполняется указанное неравенство:

$$Q_\rho(\mathfrak{D}) = \frac{1}{m} \sum_{i=1}^m \prod_{\{\mathbf{s}'_j\}_i} \prod_{\{\mathbf{s}''_k\}_i} [\rho(\mathbf{s}_i, \mathbf{s}'_j) < \rho(\mathbf{s}_i, \mathbf{s}''_k)].$$

Постановка задачи выбора функции расстояния ρ может быть сведена к задаче увеличения такого критерия качества.

Теорема 1. Функция расстояния DTW является инвариантной на классах эквивалентности.

Доказательство. Рассмотрим два объекта \mathbf{s}_1 и \mathbf{s}_2 из одного класса эквивалентности. Предположим, что один отличается от другого одним из указанных ранее способов. Для каждого из них - очевидно, что расстояние будет не более эпсилон. Так как один из другого получается путем таким трансформаций - общее расстояние будет не более эпсилон на число трансформаций. Замечательно. Предположим, что есть два объекта из разных классов эквивалентности такие, что расстояние менее эпсилон. Тогда существует путь такой, что каждая невязка меньше эпсилон. Рассмотрим такой путь. Окажется, что если он перескакивает какой-то элемент - то делаем какую-то трансформацию, и получается что есть набор операций, которые переводят один объект в другой объект. Значит, все ОК. **ДОПИСАТЬ ДОКАЗАТЕЛЬСТВО ТЕОРЕМЫ.** \square

2.3. Постановка задачи построения центроида

При решении задач метрическим методом используется матрица попарных расстояний между объектами для анализа и последующих выводов. Такой подход является вычислительно затратным, так как требует вычисления расстояний между всеми объектами выборки. Для оптимизации объема вычислений возможно определить один объект внутри класса эквивалентности, являющийся наиболее репрезентативным. Назовем его центроидом и введем формальное определение.

Определение 18. Пусть \mathfrak{D}_e — множество элементов из \mathfrak{D} , принадлежащих одному классу эквивалентности $e \in \mathbb{Y}$. Центроидом множества векторов $\mathfrak{D}_e = \{\mathbf{s}_i | y_i = e\}_{i=1}^m$ по расстоянию ρ назовем вектор $\mathbf{c}_e \in \mathbb{R}^n$ такой, что

$$\mathbf{c}_e = \operatorname{argmin}_{\mathbf{c} \in \mathbb{R}^n} \sum_{\mathbf{s}_i \in \mathfrak{D}_e} \rho(\mathbf{s}_i, \mathbf{c}).$$

В случае использования в качестве расстояния ρ расстояния DTW возможно найти центроид как решение оптимизационной задачи:

$$\mathbf{c}_e = \operatorname{argmin}_{\mathbf{c} \in \mathbb{R}^n} \sum_{\mathbf{s}_i \in \mathfrak{D}_e} \sum_{(t, t') \in \hat{\pi}_i} (\mathbf{s}_i(t') - \mathbf{c}(t))^2. \quad (2.5)$$

где $\hat{\pi}_i$ — выравнивающий путь между временным рядом \mathbf{s}_i и центроидом \mathbf{c} .

2.3.1. Решение задачи нахождения центроида методом DBA.

Теорема 2. [46]. Пусть дано множество векторов $\mathfrak{D}_e = \{\mathbf{s}_i | y_i = e\}_{i=1}^m$ одного класса, начальное приближение центроида \mathbf{c}_e и множество выравнивающих путей между каждым рядом и начальным приближением центроида $\{\tilde{\pi}_i\}_{i=1}^m$. Тогда локальный минимум задачи оптимизации (5.8) при единичном векторе весов в (5.4) (функция расстояния DTW) достигается при

$$\begin{aligned}\mathbf{c}_e(t) &= \frac{1}{N} \sum_{\mathbf{s}_i \in \mathfrak{D}_e} \sum_{t':(t,t') \in \tilde{\pi}_i} \mathbf{s}_i(t'), \\ N &= \sum_{\mathbf{s}_i \in \mathfrak{D}_e} \sum_{t':(t,t') \in \tilde{\pi}_i} 1.\end{aligned}\tag{2.6}$$

Доказательство. Для поиска центроида и решения задачи оптимизации (5.8) воспользуемся необходимым условием экстремума. Запишем частные производные функционала по $\mathbf{c}_e(t)$, $t = 1, \dots, T$, и приравняем их к нулю:

$$\frac{\partial F(\mathbf{c}_e, \mathfrak{D}_e)}{\partial \mathbf{c}_e(t)} = \sum_{\mathbf{s}_i \in \mathfrak{D}_e} \sum_{t':(t,t') \in \tilde{\pi}_i} 2(\mathbf{c}_e(t) - \mathbf{s}_i(t')) = 0.\tag{2.7}$$

Откуда и находим значение $\mathbf{c}_e(t)$, $t = 1, \dots, n$:

$$\mathbf{c}_e(t) = \frac{1}{N} \sum_{\mathbf{s}_i \in \mathfrak{D}_e} \sum_{t':(t,t') \in \tilde{\pi}_i} \mathbf{s}_i(t').\tag{2.8}$$

□

Данный метод вычисления центроида приведен в [46] и называется методом DBA. Там же приведено и его доказательство. При нахождении нового центроида множество выравнивающих рядов меняется, данную процедуру нужно проводить несколько раз, пока центроид не стабилизируется.

2.4. Постановки задач, решаемых метрическими методами

Метрические методы могут быть использованы для решения задач в условиях наличия малого числа объектов. Эти методы позволяют получить решение задач, которые могут быть заданы в виде стандартных постановок задач классификации, кластеризации и поиска оптимальной подпоследовательности. Приведем примеры таких постановок задач.

2.4.1. Задача классификации

Дана выборка $\mathfrak{D} = \{(\mathbf{s}_i, y_i)\}_{i=1}^m$, состоящая из пар объект — ответ. Объектами служат пространственно-временные объекты, а ответами являются метки класса — $y_i \in \mathbb{Y} = \{1, \dots, E\}$, где $E \ll m$. Объекты, соответствующие одной метке класса могут относиться как к одному, так и к нескольким классам эквивалентности. Объекты, соответствующие различным меткам класса, относятся к различным классам эквивалентности.

Выборка разделена на выборку для обучения $\mathfrak{D}_l = \{(\mathbf{s}_i, y_i)\}_{i=1}^{m_1}$ и выборку для контроля $\mathfrak{D}_t = \{(\mathbf{s}_i, y_i)\}_{m_1}^{m_1+m_2}$.

Пусть модель классификации f принадлежит множеству моделей метрической классификации, которые классифицируемому объекту ставят в соответствие метку класса ближайшего объекта из выборки для обучения по заданной функции расстояния ρ :

$$\hat{y} = f(\mathbf{s}'|\rho) = y \underset{i=1, \dots, m_1}{\operatorname{argmin}} \rho(s', s_i).$$

Критерий качества Q модели f для задачи классификации — доля правильно прописанного класса на выборке для контроля:

$$Q(f|\rho) = \frac{1}{m_2} \sum_{i=m_1}^{m_1+m_2} [f(\mathbf{s}_i|\rho) = y_i].$$

Требуется выбрать функцию расстояния ρ для модели классификации $f : \mathbb{R}^{n \times n} \rightarrow \mathbb{Y}$, максимизирующую критерий качества Q на контрольной выборке:

$$\rho = \underset{\rho}{\operatorname{argmax}} (Q(f|\rho)). \quad (2.9)$$

Следствие 1. Для указанной постановки задачи функция расстояния DTW является оптимальной для решения задачи классификации.

Доказательство. Надо рассмотреть пары объектов в обучающей и контрольной выборке для разных случаев: когда есть из одного класса эквивалентности и одного класса, из разных классов эквивалентности + одного класса (нет объекта в обучении из того же класса) и когда разные классы эквивалентности и разные классы. Для каждого применить историю с теоремой. \square

2.4.2. Задача кластеризации

Задана выборка $\mathfrak{D} = \{(\mathbf{s}_i)\}_{i=1}^m$, состоящая из пространственно-временных объектов. Требуется определить классы эквивалентности $y_i \in \mathbb{Y} = \{1, \dots, E\}$, где $E \ll m$.

Пусть модель кластеризации f принадлежит множеству моделей метрической кластеризации, которые относят объект к тому классу эквивалентности,

центральный объект которого является ближайшим к кластеризуемому объекту по расстоянию ρ :

$$\hat{y} = f(\mathbf{s}|\rho) = y_{\underset{i=1, \dots, E}{\operatorname{argmin}} \rho(s, c_i)},$$

где c_i — это центральный объект класса эквивалентности (или центроид).

Критерий качества Q модели f для задачи кластеризации — доля правильно проставленного класса эквивалентности на выборке:

$$Q(f|\rho) = \frac{1}{m} \sum_{i=1}^m [f(\mathbf{s}_i|\rho) = y_i].$$

СТОИТ ЛИ ТУТ ИСПОЛЬЗОВАТЬ ДРУГОЙ КРИТЕРИЙ КАЧЕСТВА, А Н СУПЕРВАЙЗД?

Требуется выбрать функцию расстояния ρ для модели кластеризации $f : \mathbb{R}^{n \times n} \rightarrow \mathbb{Y}$, максимизирующую критерий качества Q :

$$\rho = \underset{\rho}{\operatorname{argmax}}(Q(f|\rho)). \quad (2.10)$$

2.4.3. Задача поиска подпоследовательности

Рассматривается задача точной локализации подпоследовательностей. Пусть \mathbf{s} — многомерный временной ряд длины n , а \mathbf{c} — многомерный временной ряд длины $m \ll n$.

Определение 19. Оптимальная подпоследовательность в \mathbf{s} для временного ряда \mathbf{c} — это подпоследовательность $\mathbf{z} = \{\mathbf{x}_l, \dots, \mathbf{x}_{l+m}\}$ ряда \mathbf{s} , которая входит в тот же класс эквивалентности, что и временной ряд \mathbf{c} .

Из теоремы следует, что оптимальная подпоследовательность — такая, что расстояние DTW между \mathbf{z} и \mathbf{c} минимально.

Определение 20. Определим k -ую оптимальную подпоследовательность так: это подпоследовательность ряда \mathbf{s} такая, что значение расстояния DTW от нее до \mathbf{c} является с k -ой по величине.

Обозначим через \mathcal{S} множество подпоследовательностей \mathbf{s} из того же класса эквивалентности, что и \mathbf{c} . Задача поиска подпоследовательности состоит в том, чтобы определить положения всех $s \in \mathcal{S}$. Будем обозначать множество индексов, соответствующих местоположению индекса начального элемента таких подпоследовательностей во всей последовательности \mathbf{s} через $\mathbf{T} = \{t_1, \dots, t_j\}$.

Задача поиска подпоследовательности сводится к поиску индексов из множества \mathbf{T} . Функция расстояния позволяет отранжировать все возможности подпоследовательности \mathbf{s} по величине расстояния между этой подпоследовательностью и эталоном \mathbf{c} .

Считается, что t_i найдена, если одна из j первых оптимальных подпоследовательностей из упорядоченного списка $\hat{\mathcal{S}}$ пересекается с $s \in \mathcal{S}$ по длине, большей, чем 80% длины s . Таким образом, s найдена, если она достаточно сильно пересекается хотя бы с одной из предполагаемых подпоследовательностей.

ПРОПИСАТЬ ОПТИМИЗАЦИОННУЮ ЗАДАЧУ ПОИСКА РАССТОЯНИЯ

Глава 3

Обобщение матричного, многомерного и единовременного выравнивания временных рядов

В главе 2 введено обобщение метода выравнивания на объекты сложной структуры с большим количеством размерностей и осей времени. Предлагается провести анализ предложенных методов на примере трех частных случаев теоретического обобщения, а именно: наличие у объектов двух осей времени, работа с многомерными объектами и единовременная работа с несколькими временными рядами.

3.1. Матричное выравнивание

Метод динамического выравнивания двух временных рядов обобщен на случай объектов, определенных на двух осях времени. В дискретном представлении такие объекты являются матрицами. Метод динамического выравнивания временных рядов обобщается как метод динамического выравнивания матриц. Предложена функция расстояния, устойчивая к монотонным нелинейным деформациям декартового произведения двух и более временных шкал. Определен выравнивающий путь между объектами. В дальнейшем объектом называется матрица, в которой строки и столбцы соответствуют осям времени. Исследованы свойства предложенной функции расстояния. Для иллюстрации метода решаются задачи метрической классификации объектов на модельных данных и данных из датасета MNIST.

3.1.1. Постановка задачи матричного выравнивания

Рассмотрим задачу построения функции расстояния между объектами. Функция расстояния инвариантна к допустимым преобразованиям осей времени: глобальным и локальным линейным и нелинейным деформациям временной шкалы.

На двух временных осях заданы объекты вида $\mathbf{s}(t_1, t_2) \in \mathbb{R}^{n \times n}$. Функция $G_w(\mathbf{s}) : \mathbb{R}^{n \times n} \rightarrow \mathbb{R}^{\hat{n} \times \hat{n}}$ задает допустимые преобразования исходного объекта \mathbf{s} : глобальные сдвиги, локальные линейные и нелинейные деформации, а именно растяжения и сжатия оси времени, сдвиги значений по оси времени. Скалярный параметр $w \in \mathbb{R}^+$ функции G фиксирует набор этих преобразований.

Допустимым элементарным преобразованием матрицы \mathbf{s} назовем дуплицирование случайных строк и столбцов исходной матрицы, добавление или удаление крайних строк и столбцов. Допустимым преобразованием $G_w(\mathbf{s})$ является последовательность допустимых элементарных преобразований матрицы \mathbf{s} .

Функцию расстояния между двумя объектами $\rho : \mathbb{R}^{n \times n} \times \mathbb{R}^{\hat{n} \times \hat{n}} \rightarrow \mathbb{R}^+$ оценим на выборке $\mathfrak{D} = \{\mathbf{s}_i\}_{i=1}^m$ объектов вида $\mathbf{s}_i \in \mathbb{R}^{n \times n}$.

Определим функцию расстояния matrix Dynamic Time Warping (mDTW), модифицированную для случая выравнивания двойной шкалы времени. Зададим гиперпараметры алгоритма. Тензор отклонений $\Omega^{n \times n \times n \times n}$ определяется по формуле

$$\Omega(i, j, k, l) = (\mathbf{s}_1(i, j) - \mathbf{s}_2(k, l))^2.$$

Путь π между объектами $\mathbf{s}_1, \mathbf{s}_2 \in \mathbb{R}^{n \times n}$ — множество индексов тензора Ω , удовлетворяющее следующим условиям в случае матриц:

Частичный порядок. Для элементов пути π с фиксированными значениями i, k задан порядок: выравнивающий путь для фиксированных строк двух матриц упорядочен — $\{(i, j_r, k, l_r)\}_{r=1}^R \subset \pi$. Аналогично для фиксированных столбцов с индексами j, l .

Границные условия. Пусть $(i, j, k, l) \in \pi$, тогда $(1, j, 1, l) \in \pi$ и $(i, 1, k, 1) \in \pi$. Путь π содержит элементы тензора Ω : $(1, 1, 1, 1) \in \pi$ и $(n, n, n, n) \in \pi$.

Непрерывность по направлению. Для упорядоченного подмножества пути $\{(i, j_r, k, l_r)\}_{r=1}^R \subset \pi$ выполняется условие непрерывности:

$$j_r - j_{r-1} \leq 1, \quad l_r - l_{r-1} \leq 1, \quad r = 2, \dots, R.$$

В шаге пути π по фиксированному направлению времени i, k встречаются только соседние элементы матрицы (включая соседние по диагонали). Аналогично для фиксированных j, l .

Монотонность по направлению. Для упорядоченного подмножества пути $\{(i, j_r, k, l_r)\}_{r=1}^R \subset \pi$ выполняется хотя бы одно из условий монотонности функции выравнивания времени:

$$j_r - j_{r-1} \geq 1, \quad l_r - l_{r-1} \geq 1, \quad r = 2, \dots, R.$$

Условия монотонности и непрерывности можно выразить в терминах гиперпараметра алгоритма $L(i, j, k, l)$:

$$L(i, j, k, l) = \{(i-1, j, k, l), (i, j-1, k, l), (i, j, k-1, l), (i, j, k, l-1), \\ (i-1, j, k-1, l), (i, j-1, k, l-1), (i-1, j-1, k-1, l-1)\}.$$

Данное определение пути между матрицами обобщает определение пути между обычными временными рядами. Как и ранее, стоимость $\text{Cost}(\mathbf{s}_1, \mathbf{s}_2, \pi)$ пути π между объектами $\mathbf{s}_1, \mathbf{s}_2$ вычисляется по формуле

$$\text{Cost}(\mathbf{s}_1, \mathbf{s}_2, \pi) = \sum_{(i,j,k,l) \in \pi} \Omega(i, j, k, l). \quad (3.1)$$

Выравнивающий путь $\widehat{\pi}$ между объектами $\mathbf{s}_1, \mathbf{s}_2$ — путь наименьшей стоимости среди всех возможных путей между объектами:

$$\widehat{\pi} = \underset{\pi}{\operatorname{argmin}} \text{Cost}(\mathbf{s}_1, \mathbf{s}_2, \pi). \quad (3.2)$$

Функция расстояния $\rho(\mathbf{s}_1, \mathbf{s}_2) = \text{mDTW}(\mathbf{s}_1, \mathbf{s}_2)$ между объектами \mathbf{s}_1 и \mathbf{s}_2 рассчитывается как стоимость выравнивающего пути $\widehat{\boldsymbol{\pi}}$:

$$\text{mDTW}(\mathbf{s}_1, \mathbf{s}_2) = \text{Cost}(\mathbf{s}_1, \mathbf{s}_2, \widehat{\boldsymbol{\pi}}). \quad (3.3)$$

3.1.2. Алгоритм вычисления значения расстояния (3.3).

Построение алгоритма вычисления значения функции расстояния между матрицами основано на алгоритме расчета функции расстояния между обычными временными рядами.

Предлагаемый алгоритм переносит рассуждения базового алгоритма DTW на случай выравнивания двух матриц $\mathbf{s}_1, \mathbf{s}_2$. Элемент $\gamma(i, j, k, l)$ итогового тензора расстояний $\boldsymbol{\gamma}$ соответствует стоимости выравнивающего пути между $\mathbf{s}_1(1 : i, 1 : j) = \mathbf{s}_1(t_1, t_2), t_1 = 1, \dots, i, t_2 = 1, \dots, j$ и $\mathbf{s}_2(1 : k, 1 : l) = \mathbf{s}_2(t_1, t_2), t_1 = 1, \dots, k, t_2 = 1, \dots, l$. Выравнивающий путь между этими подматрицами получен одним из семи способов — если стоимость выравнивающего пути между подматрицами $\mathbf{s}_1(1 : \bar{i}, 1 : \bar{j})$ и $\mathbf{s}_2(1 : \bar{k}, 1 : \bar{l})$ минимальна для $\bar{i}, \bar{j}, \bar{k}, \bar{l}$ из множества

$$\begin{aligned} \bar{i}, \bar{j}, \bar{k}, \bar{l} \in & \{ \{i - 1, j, k, l\}, \{i, j - 1, k, l\}, \{i, j, k - 1, l\}, \{i, j, k, l - 1\}, \\ & \{i - 1, j, k - 1, l\}, \{i, j - 1, k, l - 1\}, \{i - 1, j - 1, k - 1, l - 1\} \}, \end{aligned}$$

то к выравнивающему пути между этими подматрицами добавляется элемент пути (i, j, k, l) и поправка $d\boldsymbol{\pi}$ пути $\boldsymbol{\pi}$, алгоритм вычисления которой приведен ниже.

Обозначим выравнивающий путь между $\mathbf{s}_1(1 : i, 1 : j)$ и $\mathbf{s}_2(1 : k, 1 : l)$ как $\boldsymbol{\pi}(i, j, k, l)$, тогда поправка $d\boldsymbol{\pi}$ пути $\boldsymbol{\pi}(i, j, k, l)$ при фиксированных $\bar{i}, \bar{j}, \bar{k}, \bar{l}$ вычисляется следующим образом:

Correction ($\bar{i}, \bar{j}, \bar{k}, \bar{l}, \boldsymbol{\pi}(\bar{i}, \bar{j}, \bar{k}, \bar{l})$):

if $\bar{i}, \bar{j}, \bar{k}, \bar{l} \in \{(i - 1, j, k, l), (i, j, k - 1, l), (i - 1, j, k - 1, l)\}$:

$\boldsymbol{\pi} = \{(\bar{i}, r, \bar{k}, f) \in \boldsymbol{\pi}(\bar{i}, \bar{j}, \bar{k}, \bar{l}) \mid r, f \in \mathbb{N}\}$

elif $\bar{i}, \bar{j}, \bar{k}, \bar{l} \in \{(i, j - 1, k, l), (i, j, k, l - 1), (i, j - 1, k, l - 1)\}$:

$\boldsymbol{\pi} = \{(r, \bar{j}, f, \bar{l}) \in \boldsymbol{\pi}(\bar{i}, \bar{j}, \bar{k}, \bar{l}) \mid r, f \in \mathbb{N}\}$

elif $\bar{i}, \bar{j}, \bar{k}, \bar{l} = (i - 1, j - 1, k - 1, l - 1)$:

$\boldsymbol{\pi} = \{(\bar{i}, r, \bar{k}, f) \in \boldsymbol{\pi}(\bar{i}, \bar{j}, \bar{k}, \bar{l}) \mid r, f \in \mathbb{N}\} \cup \{(r, \bar{j}, f, \bar{l}) \in \widehat{\boldsymbol{\pi}}(\bar{i}, \bar{j}, \bar{k}, \bar{l}) \mid r, f \in \mathbb{N}\}$

$d\boldsymbol{\pi} = \{ \text{element} \in \boldsymbol{\pi} : \text{произведены замены индексов } \bar{i} = i, \bar{j} = j, \bar{k} = k, \bar{l} = l \}$

return $d\boldsymbol{\pi}$

Рис. 3.1. Алгоритм вычисления поправки $d\boldsymbol{\pi}$ пути $\boldsymbol{\pi}$.

Алгоритм динамического выравнивания двух матриц и вычисления расстояния mDTW между ними с учетом приведенного выше алгоритма примет вид:

$mDTW(s_1, s_2)$:

$$\begin{aligned} \gamma(1:n+1, 1:n+1, 1:n+1, 1:n+1) &= \inf \\ \gamma(1, 1, 1, 1) &= 0 \\ \pi(1, 1, 1, 1) &= ((1, 1), (1, 1)) \\ \text{for } i, j, k, l \in \mathbb{N}^{2:n+1} \times \mathbb{N}^{2:n+1} \times \mathbb{N}^{2:n+1} \times \mathbb{N}^{2:n+1}: \\ \bar{i}, \bar{j}, \bar{k}, \bar{l} &= \arg \min ([\gamma(i-1, j, k, l), \gamma(i, j-1, k, l), \gamma(i, j, k-1, l), \\ &\quad \gamma(i, j, k, l-1), \gamma(i-1, j, k-1, l), \gamma(i, j-1, k, l-1), \\ &\quad \gamma(i-1, j-1, k-1, l-1)]) \\ d\pi &= \text{Correction}(\bar{i}, \bar{j}, \bar{k}, \bar{l}, \pi(\bar{i}, \bar{j}, \bar{k}, \bar{l})) \\ \pi(i, j, k, l) &= d\pi \cup \{(\bar{i}, \bar{j}, \bar{k}, \bar{l})\} \\ cost &= (s_1(i, j) - s_2(k, l))^2 + \sum_{(r, f, t, g) \in d\pi} (s_1(r, f) - s_2(t, g))^2 \\ \gamma(i, j, k, l) &= cost + \gamma(\bar{i}, \bar{j}, \bar{k}, \bar{l}) \\ \text{return } &\sqrt{\gamma(n+1, n+1, n+1, n+1)} \end{aligned}$$

Рис. 3.2. Алгоритм вычисления расстояния между матрицами.

Следует отметить, что алгоритм [5] имеет высокую сложность вычисления — $O(n^4)$. Использование Sakoe-Chiba band сокращает вычислительную сложность до $O(n^2k^2)$, где k — параметр ограничения Sakoe-Chiba band.

3.2. Выравнивание многомерных временных рядов

Предлагается обобщить метод динамического выравнивания двух одномерных временных рядов на случай многомерных объектов. В дискретном представлении такие объекты являются матрицами: число измерений на ось времени. Метод динамического выравнивания временных рядов обобщается как метод динамического выравнивания матриц с фиксированными временными измерениями. Предложена функция расстояния, устойчивая к монотонным нелинейным деформациям. Определен выравнивающий путь между объектами. В дальнейшем объектом называется матрица, в которой строки соответствуют различным измерениям, а столбцы соответствуют отсчетам времени. Исследованы свойства предложенной функции расстояния.

3.2.1. Постановка задачи многомерного выравнивания

Рассмотрим задачу построения функции расстояния между объектами. Функция расстояния инвариантна к допустимым преобразованиям осей времени: глобальным и локальным линейным и нелинейным деформациям временной шкалы.

Заданы объекты вида $\mathbf{S} \in \mathbb{R}^{n \times m}$. Такие объекты возможно представить в виде вектора $(\mathbf{S}(1), \dots, \mathbf{S}(n))$, в котором каждый элемент $\mathbf{s}(i)$ является скаляром

ным вектором размерности m : $\mathbf{s}(i) \in \mathbb{R}^m$. Функция $G_w(\mathbf{S}) : \mathbb{R}^{n \times m} \rightarrow \mathbb{R}^{\hat{n} \times \hat{m}}$ задает допустимые преобразования исходного объекта \mathbf{S} : глобальные сдвиги, локальные линейные и нелинейные деформации, а именно растяжения и сжатия оси времени, сдвиги значений по оси времени. Скалярный параметр $w \in \mathbb{R}^+$ функции G фиксирует набор этих преобразований.

Допустимым элементарным преобразованием вектора \mathbf{S} назовем дуплицирование и удаление случайных элементов вектора \mathbf{S} и добавление случайного шума к векторам \mathbf{s}_i . Допустимым преобразованием $G_w(\mathbf{S})$ является последовательность допустимых элементарных преобразований \mathbf{S} .

Функцию расстояния между двумя объектами $\rho : \mathbb{R}^{n \times m} \times \mathbb{R}^{\hat{n} \times \hat{m}} \rightarrow \mathbb{R}^+$ оценим на выборке $\mathfrak{D} = \{\mathbf{S}_i\}_{i=1}^m$ объектов вида $\mathbf{S}_i \in \mathbb{R}^{n \times m}$. Предполагается, что решается задача метрического анализа данных с заранее заданным критерием качества Q .

Определим тензор отклонений Ω для многомерного случая. При использовании функции расстояния между двумя объектами \mathbf{S}_1 и \mathbf{S}_2 , основанной на методе динамического выравнивания осей времени, возникает возможность варьировать параметр ϕ алгоритма a поиска оптимального пути $\hat{\pi}$.

$$\phi^* = \arg \max_{\phi \in \Phi} Q(f|\rho), \quad (3.4)$$

где Φ — это множество возможных параметров. ρ — функция расстояния, которая вычисляется на основе результатов работы алгоритма поиска выравнивающего пути a .

В качестве множества Φ возможно взять множество $\Phi = \{\phi_{L_1}, \phi_{L_2}, \phi_{\text{cosine}}\}$, где значения между элементами \mathbf{s}_t , \mathbf{c}_τ временных рядов \mathbf{s} и \mathbf{c} вычисляются как:

- 1) $\phi_{L_1}(\mathbf{s}_t, \mathbf{c}_\tau) = \|\mathbf{s}_t - \mathbf{c}_\tau\|_1$;
- 2) $\phi_{L_2}(\mathbf{s}_t, \mathbf{c}_\tau) = \|\mathbf{s}_t - \mathbf{c}_\tau\|_2$;
- 3) $\phi_{\text{cosine}}(\mathbf{s}_t, \mathbf{c}_\tau) = 1 - \frac{\mathbf{s}_t^\top \mathbf{c}_\tau}{\|\mathbf{s}_t\|_2 \|\mathbf{c}_\tau\|_2}$.

К этому множеству возможно добавить произвольную метрику между двумя скалярными векторами.

Таким образом, тензор отклонений для многомерного случая можно вычислить как:

$$\Omega(i, j) = \phi(\mathbf{S}_1(i), \mathbf{S}_2(j)).$$

Путь π между объектами $\mathbf{S}_1, \mathbf{S}_2 \in \mathbb{R}^{n \times m}$ — множество индексов тензора Ω , удовлетворяющее следующим условиям:

Частичный порядок. Для элементов пути π с фиксированными значениями i, k задан порядок: выравнивающий путь для фиксированных строк двух матриц упорядочен — $\{(i_r, j_r)\}_{r=1}^R \subset \pi$.

Границные условия. Пусть $(i, j) \in \pi$, тогда $(1, j) \in \pi$ и $(i, 1) \in \pi$. Путь π содержит элементы тензора Ω : $(1, 1) \in \pi$ и $(n, n) \in \pi$.

Непрерывность по направлению. Для упорядоченного подмножества пути $\{(i_r, j_r)\}_{r=1}^R \subset \pi$ выполняется условие непрерывности:

$$i_r - i_{r-1} \leq 1, \quad j_r - j_{r-1} \leq 1, \quad r = 2, \dots, R.$$

В шаге пути π по каждому фиксированному направлению времени i, k встречаются только соседние элементы матрицы (включая соседние по диагонали).

Монотонность по направлению. Для упорядоченного подмножества пути $\{(i_r, j_r)\}_{r=1}^R \subset \pi$ выполняется хотя бы одно из условий монотонности функции выравнивания времени:

$$i_r - i_{r-1} \geq 1, \quad j_r - j_{r-1} \geq 1, \quad r = 2, \dots, R.$$

Кроме того, предполагается использование Sakoe-Chiba band ширины k , где k — некоторый параметр.

Аналогично двумерным временным рядам, условия монотонности и непрерывности также выражаются через гиперпараметр алгоритма $L(i, j)$:

$$L(i, j) = \{(i-1, j), (i, j-1), (i-1, j-1)\}.$$

Наконец, зададим для многомерных временных рядов S_1 и S_2 стоимость $\text{Cost}(S_1, S_2, \pi)$ пути

$$\text{Cost}(S_1, S_2, \pi) = \sum_{(i,j) \in \pi} \Omega(i, j). \quad (3.5)$$

Обобщим выведенный ранее алгоритм выравнивания на случай с многомерными тензорами S_1, S_2 . Аналогично двумерному случаю, посчитаем тензор расстояний γ , элемент $\gamma(i, j)$ которого характеризует стоимость выравнивающего пути $S_1(1 : i)$ и $S_2(1 : j, 1 : l)$, где выравнивающий путь найден таким образом, чтобы его стоимость была минимальна для \bar{i}, \bar{j} из множества

$$\bar{i}, \bar{j} \in \{\{i-1, j\}, \{i, j-1\}, \{i-1, j-1\}\}.$$

Найденный выравнивающий путь для указанных выше подматриц будем обозначать $\pi(i, j)$.

3.2.2. Алгоритм вычисления многомерного расстояния DTW

ТУТ БУДЕТ АЛГОРИТМ

3.3. Одновременное выравнивание нескольких временных рядов

Предлагается обобщить метод динамического выравнивания двух временных рядов на случай трех объектов. Метод динамического выравнивания временных рядов обобщается как метод динамического выравнивания набора временных рядом. Предложена функция расстояния, устойчивая к монотонным

нелинейным деформациям. Определен выравнивающий путь между объектами. Исследованы свойства предложенной функции расстояния.

Заданы объекты вида $\mathbf{s} \in \mathbb{R}^n$. Такие объекты возможно представить в виде вектора (s_1, \dots, s_n) , в котором каждый элемент s_i является скаляром. Функция $G_w(\mathbf{s}) : \mathbb{R}^n \rightarrow \mathbb{R}^{\hat{n}}$ задает допустимые преобразования исходного объекта \mathbf{s} : глобальные сдвиги, локальные линейные и нелинейные деформации, а именно растяжения и сжатия оси времени, сдвиги значений по оси времени. Скалярный параметр $w \in \mathbb{R}^+$ функции G фиксирует набор этих преобразований.

Очевидно, что для трех объектов $\mathbf{s}_1(t)$, $\mathbf{s}_2(t)$ и $\mathbf{s}_3(t)$ из одного класса эквивалентности выполняется следующее условие: $w_1, w_2, w_3 : G_{w_1}(\mathbf{s}_1)$

Необходимо построить алгоритм выравнивания между этими объектами и доопределить понятие расстояния и пути в такой постановке.

Индексами всех возможных невязок в таком случае является $\mathcal{I}_{\mathbf{s}_1} \times \mathcal{I}_{\mathbf{s}_2} \times \mathcal{I}_{\mathbf{s}_3} = \{1, \dots, n\} \times \{1, \dots, n\} \times \{1, \dots, n\} = \{\{1, 1, 1\}, \{1, 1, 2\}, \dots, \{n, n, n\}\}$. Допустимое множество индексов невязок для простоты примем равным всем возможным индексам невязок: $\mathbb{I} = \mathcal{I}_{\mathbf{s}_1} \times \mathcal{I}_{\mathbf{s}_2} \times \mathcal{I}_{\mathbf{s}_3}$. Необходимо определить и функцию ϕ , которая в настоящем примере должна быть определена на трех объектах. С ее помощью получится вычислить тензор отклонений Ω следующим образом: для всех $\mathbf{l} \in \mathbb{I} : \Omega_{\mathbf{l}} = \phi(\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3, \mathbf{l})$.

Можем, например, определить, как среднее суммы попарных расстояний:

$$\phi(\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3, \{i, j, k\}) = \frac{1}{3} [|s_{1i} - s_{2j}| + |s_{1i} - s_{3k}| + |s_{2j} - s_{3k}|].$$

Определим и множество элементов L , предшествующих элементу $\{i, j, k\}$:

$$\begin{aligned} L(\{i, j, k\}) &= \{\{i-1, j, k\}, \{i, j-1, k\}, \{i, j, k-1\}, \\ &\quad \{i-1, j-1, k\}, \{i-1, j, k-1\}, \{i, j-1, k-1\}, \{i-1, j-1, k-1\}\} \end{aligned}$$

Определим новый тензор Γ , размерность которого соответствует тензору Ω . Для вычисления значений нового тензора используются следующие формулы:

$$\begin{aligned} \Gamma_{1,1,k} &= \Omega_{1,1,K}, \Gamma_{1,j,1} = \Omega_{1,j,1}, \Gamma_{i,1,1} = \Omega_{i,1,1}; \\ &\text{Для всех } \mathbf{l} \in \mathbb{I} \\ \Gamma(\mathbf{l}) &= \Omega_{\mathbf{l}} + \min_{\mathbf{i} \in L(\mathbf{l})} \{\Gamma_{\mathbf{i}}\}. \end{aligned}$$

Примем $\Gamma_{n,n,n}$ за расстояние между объектами.

3.3.1. Алгоритм единовременного вычисления расстояния DTW

ТУТ БУДЕТ АЛГОРИТМ

Глава 4

Обобщение метода выравнивания в непрерывном пространстве

Задача мониторинга физической активности человека с помощью носимых приборов требует значительных ограничений на скорость работы алгоритма и количество доступной ему памяти. Для того чтобы удовлетворять этим ограничениям, предлагается перейти от дискретного представления временных рядов к их аналитическому описанию и анализировать их, используя математические модели.

4.1. Непрерывное выравнивание временных рядов

В работе автора [4] рассматривается задача классификации физической активности человека. Используется метрический алгоритм классификации, причем класс объекта определяется по методу ближайшего центроида. Предлагается аппроксимировать все временные ряды с помощью сплайнов и вычислять расстояния до центроидов с использованием непрерывного выравнивающего пути. Вычисление расстояния производится с помощью аналитических преобразований.

4.2. Переход от дискретных объектов к непрерывным

Дискретный временной ряд \mathbf{s} — это упорядоченная во времени последовательность. Введем аналогичное определение для случая непрерывного времени.

Определение 21. Непрерывный временной ряд на интервале времени $\widehat{T} = [0; T]$ — это непрерывная функция $s^c(t) : \widehat{T} \rightarrow \mathbb{R}$.

Пусть \mathcal{S} — пространство всех дискретных временных рядов, а \mathcal{S}^c — пространство всех непрерывных временных рядов. Каждый дискретный объект $s \in \mathcal{S}$ ассоциирован со своим непрерывным аналогом $s^c(t) \in \mathcal{S}^c$.

Интерполяция дискретных временных рядов или их аппроксимация параметрическими функциями определяет отображение $f : \mathcal{S} \rightarrow \mathcal{S}^c$. Алгоритм состоит из трех основных частей: выбор семейства параметрических функций, решение задачи оптимизации для поиска оптимальных параметров и построение аппроксимации. Мы будем использовать приближение кубическими сплайнами [56].

Приближение кубическими сплайнами

Множество узлов $\{(x_i, y_i)\}_{i=1}^n$ в \mathbb{R}^2 порождается гладкой функцией $f(x)$. В нашем случае x обозначает время, а y — некоторое измерение. Кубический многочлен приближает функцию между двумя соседними узлами: (x_i, y_i) и (x_{i+1}, y_{i+1}) . На границе значения соседних многочленов, а также их первые и

вторые производные, совпадают. Данное множество точек $\{(x_i, y_i)\}_{i=1}^n$ называется узлами сплайна.

4.3. Расстояние DTW на непрерывной оси времени

Стандартный алгоритм DTW определяет расстояние между дискретными объектами. Определим понятия пути, его стоимости и расстояния DTW для непрерывного случая.

Рассмотрим два дискретных временных ряда s_1 и s_2 , а также их непрерывные аналоги, две функции: $s_1^c(t_1)$ и $s_2^c(t_2)$, $t_1, t_2 \in [0, T]$.

Определение 22. Путь π^c между двумя непрерывными временными рядами s_1 и s_2 — это возрастающая непрерывная функция $\pi^c : t_1 \mapsto t_2$, удовлетворяющая граничным условиям

$$\begin{aligned}\pi^c &\in C_{[0;T]}, \quad \pi^c(0) = 0, \quad \pi^c(T_1) = T_2. \\ t_1 > t'_1 &\Rightarrow \pi^c(t_1) > \pi^c(t'_1).\end{aligned}$$

Определение 23. Стоимость $\text{Cost}(s_1^c(t_1), s_2^c(t_2), \pi^c)$ пути π^c между двумя непрерывными временными рядами $s_1^c(t_1)$ и $s_2^c(t_2)$ определяется по формуле

$$\text{Cost}(s_1^c(t_1), s_2^c(t_2), \pi^c) = \frac{1}{L} \int_{t_1} |s_1^c(t_1) - s_2^c(\pi^c(t_1))| dt_1,$$

где L — длина кривой, являющейся графиком функции $\pi^c(t)$, $t \in [0, T_1]$.

Определение 24. Выравнивающий путь $\hat{\pi}^c$ между двумя непрерывными временными рядами $s_1^c(t_1)$ и $s_2^c(t_2)$ — это функция $\hat{\pi}^c$ с наименьшей стоимостью

$$\hat{\pi}^c = \underset{\pi^c}{\operatorname{argmin}} \text{Cost}(s_1^c(t_1), s_2^c(t_2), \pi^c).$$

Определение 25. DTW расстояние между двумя непрерывными временными рядами — это стоимость выравнивающего пути

$$\text{DTW}(s_1^c(t_1), s_2^c(t_2)) = \text{Cost}(s_1^c(t_1), s_2^c(t_2), \hat{\pi}^c).$$

4.4. Выравнивающий путь и его свойства в случае непрерывной оси времени

Сформулируем две леммы, справедливые для случая непрерывных временных рядов.

Лемма 1. Пусть $s_1(t)$ и $s_2(t)$ — два временных ряда с константой Липшица L , $\hat{\pi}^c : t_1 \mapsto t_2$ — выравнивающий путь между ними. Тогда при малых изменениях этого пути его стоимость не меняется сильно:

$$\|\hat{\pi}^c - \pi^c\|_C \leq \varepsilon \quad \Rightarrow \quad |\text{Cost}(s_1, s_2, \hat{\pi}^c) - \text{Cost}(s_1, s_2, \pi^c)| \leq \varepsilon TL.$$

Здесь T задает границу временного интервала, на котором рассматриваются ряды, $\varepsilon > 0$.

Доказательство. Запишем цепочку неравенств, которая доказывает наше утверждение

$$\begin{aligned} & |Cost(s_1, s_2, \hat{\pi}^c) - Cost(s_1, s_2, \pi^c)| = \\ & \left| \int_{t_1} (\|s_1(t_1) - s_2(\hat{\pi}^c(t_1))\| dt_1 - \int_{t_1} (\|s_1(t_1) - s_2(\pi^c(t_1))\| dt_1) \right| \leqslant \\ & \int_{t_1} |s_2(\hat{\pi}^c(t_1)) - s_2(\pi^c(t_1))| dt_1 \leqslant \int_{t_1} L |\hat{\pi}^c(t_1) - \pi^c(t_1)| dt_1 \leqslant \\ & L \int_{t_1} \max_{t_1 \in [0, T]} (\hat{\pi}^c(t_1) - \pi^c(t_1)) dt_1 \leqslant L \int_{t_1} \|\hat{\pi}^c - \pi^c\|_C dt_1 \leqslant TL\varepsilon. \end{aligned}$$

□

Лемма 2. Пусть $s_1(t)$ и $s_2(t)$ — два временных ряда с константой Липшица L , $\hat{\pi}^c : t_1 \mapsto t_2$ — выравнивающий путь между ними. Тогда его цена не меняется сильно при малом изменении одного из временных рядов:

$$\|\hat{s}_2 - s_2\|_C \leqslant \varepsilon \Rightarrow |Cost(s_1, \hat{s}_2, \hat{\pi}^c) - Cost(s_1, s_2, \hat{\pi}^c)| \leqslant \varepsilon TL,$$

где T — граница временного интервала, на котором рассматриваются ряды.

Доказательство. Следующая цепочка неравенств доказывает требуемое утверждение:

$$\begin{aligned} & |Cost(s_1, \hat{s}_2, \hat{\pi}^c) - Cost(s_1, s_2, \hat{\pi}^c)| = \\ & \left| \int_{t_1} (\|s_1(t_1) - \hat{s}_2(\hat{\pi}^c(t_1))\| dt_1 - \int_{t_1} (\|s_1(t_1) - s_2(\hat{\pi}^c(t_1))\| dt_1) \right| \leqslant \\ & \int_{t_1} |\hat{s}_2(\hat{\pi}^c(t_1)) - s_2(\hat{\pi}^c(t_1))| dt_1 \leqslant \int_{t_1} \max_{t_1 \in [0, T]} (\hat{s}_2(\hat{\pi}^c(t_1)) - s_2(\hat{\pi}^c(t_1))) dt_1 \leqslant \\ & \int_{t_1} \|\hat{s}_2 - s_2\|_C dt_1 \leqslant T\varepsilon. \end{aligned}$$

□

В последних двух леммах продемонстрированы свойства устойчивости стоимости выравнивающего пути при малых изменениях в исходных данных или выравнивающем пути. В данной работе также вводится предположение об устойчивости выравнивающего пути при малых изменениях исходных объектов.

Предположение 1. Пусть $s_1(t)$ и $s_2(t)$ — два временных ряда с константой Липшица L , $\hat{s}_2(t)$ получен малым изменением $s_2(t)$. Тогда

$$\forall \varepsilon_1 > 0 \quad \exists \varepsilon_2(\varepsilon_1), \quad \forall \hat{s}_2(t) : \quad \|\hat{s}_2(t) - s_2(t)\|_C \leqslant \varepsilon_2 \Rightarrow \|\pi^c - \hat{\pi}^c\|_C \leqslant \varepsilon_1,$$

где π^c и $\hat{\pi}^c$ — выравнивающие пути между $s_1(t)$, $s_2(t)$ и $s_1(t)$, $\hat{s}_2(t)$ соответственно.

4.5. Вычисление выравнивающего пути

Функция $\hat{\pi}^c : t_1 \mapsto t_2$ является решением задачи оптимизации из определения выравнивающего пути. Алгоритм построения этой функции в дискретном случае использует динамическое программирование. Этот подход неприменим в случае непрерывных объектов.

Ограничим пространство, в котором будем искать $\hat{\pi}^c$ некоторым пространством параметрических функций. В таком пространстве каждый набор параметров задает ровно одну функцию. Рассмотрим функцию $\tilde{\hat{\pi}}^c$ — аппроксимацию выравнивающего пути $\hat{\pi}^c$. Если это приближение хорошее:

$$\|\tilde{\hat{\pi}}^c - \hat{\pi}^c\|_C \leq \varepsilon,$$

то его стоимость не будет сильно отличаться от стоимости истинного выравнивающего пути согласно лемме 1. Примем такое приближение за выравнивающий путь. Таким образом, задача поиска в исходном пространстве сведена к задаче оптимизации параметров $\boldsymbol{\theta}$:

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \operatorname{Cost}(s_1, s_2, \boldsymbol{\theta}) = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \int_{t_1} |s_1(t_1) - s_2(F(\boldsymbol{\theta})(t_1))| dt_1,$$

где $F(\boldsymbol{\theta})$ — отображение, которое набору параметров сопоставляет параметрическую функцию.

В данной работе предлагается аппроксимировать выравнивающий путь с помощью кубических сплайнов. Количество узлов и их координаты (x, y) могут варьироваться. Пусть N — количество узлов и пусть их координаты по оси t_1 также зафиксированы. Будем предполагать, что координаты узлов по оси t_2 $\boldsymbol{\theta} = \{t_{2i}\}_{i=1}^N$ — оптимизируемые параметры.

Свойство непрерывности для аппроксимации пути выполнено. Введем ограничения на оптимизируемые параметры, для того чтобы были выполнены свойство монотонности и граничные условия. Пусть

$$\begin{aligned} t_{21} &= t_{11}, & t_{2N} &= t_{1N}, \\ t_{2(i)} &\leq t_{2(i+1)}, & i &\in \{1, \dots, N-1\}. \end{aligned}$$

Сформулируем лемму для рассматриваемого способа аппроксимации.

Лемма 3. Стоимость выравнивающего пути не меняется сильно при малом изменении вектора параметров, которые определяют приближение пути:

$$\|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}\|_2 \leq \varepsilon \Rightarrow |\operatorname{Cost}(s_1, \hat{s}_2, F(\hat{\boldsymbol{\theta}})) - \operatorname{Cost}(s_1, s_2, F(\boldsymbol{\theta}))| \leq \delta.$$

Доказательство. Если координата y_i i -ого узла сплайна меняется не сильно, то выравнивающий путь между двумя временными рядами сильно не изменится по норме $\|\cdot\|_C$. Тогда наше утверждение следует из леммы 1. \square

Глава 5

Взвешенное выравнивание, анализ отклонений и выравнивание скрытых представлений как следствия обобщения метода выравнивания

При анализе свойств функции выравнивания были найдены способы ввести дополнительные эвристики, которые повысят точность используемых методов и могут быть теоретически обобщены на общий случай.

5.1. Взвешенное выравнивание

Метрические методы используют матрицу попарных расстояний, строящуюся при помощи фиксированной функции расстояния. Вычислительная сложность алгоритмов, использующих такую матрицу, по меньшей мере квадратична относительно числа временных рядов. Проблема снижения вычислительной сложности решается путем предварительного выделения эталонных объектов, центроидов классов, и последующего их использования для описания классов. В качестве базовой модели классификации выбрана модель, использующая динамическое выравнивание временных рядов для построения центроида.

При анализе временных рядов определенные измерения намного важнее для анализа. Предлагается ввести функцию весов центроида, влияющую на вычисление расстояния между объектами. Для анализа алгоритма построения центроида использованы как временные ряды элементарных функций, так и временные ряды физической активности человека с акселерометра мобильного телефона. Свойства построенной модели исследуются и сравниваются со свойствами модели, выбранной в качестве базовой.

5.1.1. Постановка задачи поиска оптимального метрического классификатора

Предположим, что решается задача метрической классификации при помощи модели классификации f . В данной главе параметрами модели примем множество центроидов $\mathbf{C} = \{\mathbf{c}_e\}_{e=1}^E$ и множество векторов весов центроидов $\hat{\mathbf{W}} = \{\mathbf{w}_e\}_{e=1}^E$ или же матриц весов центроидов $\hat{\mathbf{W}} = \{\mathbf{W}_e\}_{e=1}^E$.

Функцией ошибки Q модели f для задачи классификации будем считать

$$Q(f, \mathfrak{D}_t) = \frac{1}{|\mathfrak{D}_t|} \sum_{i=1}^{|\mathfrak{D}_t|} [f(\mathbf{s}_i) \neq y_i].$$

Требуется построить модель классификации $f : \mathbb{R}^n \rightarrow \mathbb{Y}$, минимизирующую функцию ошибки Q на контрольной выборке:

$$f_{\mathbf{C}, \hat{\mathbf{W}}} = \underset{\mathbf{C}, \hat{\mathbf{W}}}{\operatorname{argmin}} (Q(f, \mathfrak{D}_t)). \quad (5.1)$$

5.1.2. Вычисление значения функции расстояния

В данной работе в качестве метрического расстояния между объектами предлагается использовать *взвешенную* стоимость оптимального пути между этими объектами.

Определим соответствующую модификацию базового алгоритма DTW. Пусть даны два временных ряда: \mathbf{s}_1 и \mathbf{s}_2 . Будем считать, что $\mathbf{s}_1, \mathbf{s}_2 \in \mathbb{R}^n$. Зададим гиперпараметры алгоритма.

Пусть $\phi(\mathbf{s}_1, \mathbf{s}_2, (i, j)) = (s_{1i} - s_{2j})^2$. Путь определяется аналогично базовому алгоритму DTW. Он должен удовлетворять граничным условиям, а также условиям непрерывности и монотонности. Последние два условия равносильны выбору гиперпараметра $L(i, j) = \{(i-1, j), (i, j-1), (i-1, j-1)\}$. Кроме того, используется Sakoe-Chiba band некоторой ширины k .

Векторно-взвешенный путь наименьшей стоимости. Дадим определение векторно-взвешенного пути наименьшей стоимости, vwDTW. Дан вектор весов $\mathbf{w} \in \mathbb{R}^n$.

Определение 26. Векторно-взвешенной с весом \mathbf{w} стоимостью $\text{Cost}(\mathbf{s}_1, \mathbf{s}_2, \mathbf{w}, \boldsymbol{\pi})$ пути $\boldsymbol{\pi}$ между последовательностями \mathbf{s}_1 и \mathbf{s}_2 назовем

$$\text{Cost}(\mathbf{s}_1, \mathbf{s}_2, \mathbf{w}, \boldsymbol{\pi}) = \sum_{(i,j) \in \boldsymbol{\pi}} w_j \Omega_{ij}. \quad (5.2)$$

Определение 27. Векторно-взвешенным путем наименьшей стоимости (векторно-взвешенным выравнивающим путем) $\hat{\boldsymbol{\pi}}$ между последовательностями \mathbf{s}_1 и \mathbf{s}_2 назовем путь, имеющий наименьшую взвешенную стоимость среди всех возможных путей между последовательностями \mathbf{s}_1 и \mathbf{s}_2 :

$$\hat{\boldsymbol{\pi}} = \underset{\boldsymbol{\pi}}{\operatorname{argmin}} \text{Cost}(\mathbf{s}_1, \mathbf{s}_2, \mathbf{w}, \boldsymbol{\pi}). \quad (5.3)$$

Обозначим стоимость векторно-взвешенного выравнивающего пути между последовательностями \mathbf{s}_1 и \mathbf{s}_2 через $\rho(\mathbf{s}_1, \mathbf{s}_2, \mathbf{w}) = \text{Cost}(\mathbf{s}_1, \mathbf{s}_2, \mathbf{w}, \hat{\boldsymbol{\pi}})$.

Для вычисления стоимости такого пути в данной работе используется модифицированный метод DTW — vwDTW (vector-weighted DTW). Согласно этому методу необходимо построить новую матрицу $\boldsymbol{\gamma}$, элементы которой определяются следующим образом:

$$\begin{aligned} \gamma_{1j} &= w_j \Omega_{1j}, \quad \gamma_{i1} = w_1 \Omega_{i1}, \quad i, j = 1, \dots, n, \\ \gamma_{ij} &= w_j \Omega_{ij} + \min(\gamma_{i,j-1}, \gamma_{i-1,j}, \gamma_{i-1,j-1}), \quad i, j = 2, \dots, n. \end{aligned}$$

Элемент γ_{ij} матрицы $\boldsymbol{\gamma}$ равен стоимости векторно-взвешенного выравнивающего пути между последовательностями $\{s_{1a}\}_{a=1}^i$ и $\{s_{2a}\}_{a=1}^j$.

В качестве значения функции расстояния между двумя объектами выберем стоимость векторно-взвешенного выравнивающего пути между ними (5.3):

$$\rho(\mathbf{s}_1, \mathbf{s}_2, \mathbf{w}) = \gamma_{nn}. \quad (5.4)$$

Заметим, что при единичном векторе весов vwDTW эквивалентен обычному DTW, описание которого приведено выше.

Матрично-взвешенный путь наименьшей стоимости. Дадим определение матрично-взвешенного пути наименьшей стоимости, mwDTW. Данна матрица весов $\mathbf{W} \in \mathbb{R}^{n \times n}$.

Определение 28. Матрично-взвешенной с весом \mathbf{W} стоимостью $\text{Cost}(\mathbf{s}_1, \mathbf{s}_2, \mathbf{W}, \boldsymbol{\pi})$ пути $\boldsymbol{\pi}$ между последовательностями \mathbf{s}_1 и \mathbf{s}_2 назовем

$$\text{Cost}(\mathbf{s}_1, \mathbf{s}_2, \mathbf{W}, \boldsymbol{\pi}) = \sum_{(i,j) \in \boldsymbol{\pi}} W_{ij} \Omega_{ij}. \quad (5.5)$$

Определение 29. Матрично-взвешенным путем наименьшей стоимости (матрично-взвешенным выравнивающим путем) $\widehat{\boldsymbol{\pi}}$ между последовательностями \mathbf{s}_1 и \mathbf{s}_2 назовем путь, имеющий наименьшую матрично-взвешенную стоимость среди всех возможных путей между последовательностями \mathbf{s}_1 и \mathbf{s}_2 :

$$\widehat{\boldsymbol{\pi}} = \underset{\boldsymbol{\pi}}{\operatorname{argmin}} \text{Cost}(\mathbf{s}_1, \mathbf{s}_2, \mathbf{W}, \boldsymbol{\pi}). \quad (5.6)$$

Обозначим стоимость матрично-взвешенного выравнивающего пути между последовательностями \mathbf{s}_1 и \mathbf{s}_2 через $\rho(\mathbf{s}_1, \mathbf{s}_2, \mathbf{W}) = \text{Cost}(\mathbf{s}_1, \mathbf{s}_2, \mathbf{W}, \widehat{\boldsymbol{\pi}})$.

Вычисление его стоимости происходит с помощью еще одного модифицированного метода DTW — mwDTW (matrix-weighted DTW). Согласно этому методу необходимо построить новую матрицу $\boldsymbol{\gamma}$, элементы которой определяются следующим образом:

$$\gamma_{1j} = W_{1j} \Omega_{1j}, \quad \gamma_{i1} = W_{i1} \Omega_{i1}, \quad i, j = 1, \dots, n,$$

$$\gamma_{ij} = W_{ij} \Omega_{ij} + \min(\gamma_{i,j-1}, \gamma_{i-1,j}, \gamma_{i-1,j-1}), \quad i, j = 2, \dots, n.$$

Элемент γ_{ij} матрицы $\boldsymbol{\gamma}$ равен стоимости матрично-взвешенного выравнивающего пути между последовательностями $\{s_{1a}\}_{a=1}^i$ и $\{s_{2a}\}_{a=1}^j$.

В качестве значения функции расстояния между двумя объектами выберем стоимость матрично-взвешенного выравнивающего пути между ними (5.6):

$$\rho(\mathbf{s}_1, \mathbf{s}_2, \mathbf{W}) = \gamma_{nn}. \quad (5.7)$$

Заметим, что при использовании матрицы, состоящей из одних единиц, mwDTW переходит в обычный DTW.

5.1.3. Вычисление параметров модели классификатора

Построение центроида. Пусть множество весов $\hat{\mathbf{W}}$ фиксировано. Построим множество центроидов \mathbf{C} .

5.1.4. Постановка задачи построения центроида

Определение 30. Пусть \mathfrak{D}_e — множество элементов из \mathfrak{D} , принадлежащих одному классу e из \mathbb{Y} . Центроидом множества векторов $\mathfrak{D}_e = \{\mathbf{s}_i | y_i = e\}_{i=1}^m$ по расстоянию ρ назовем вектор $\mathbf{c}_e \in \mathbb{R}^n$ такой, что

$$\mathbf{c}_e = \operatorname{argmin}_{\mathbf{c} \in \mathbb{R}^n} \sum_{\mathbf{s}_i \in \mathfrak{D}_e} \rho(\mathbf{s}_i, \mathbf{c}),$$

где ρ — стоимость векторно- (матрично-) взвешенного пути наименьшей стоимости vwDTW (mvDTW).

Центройд найдем как решение оптимизационной задачи для vwDTW

$$\mathbf{c}_e = \operatorname{argmin}_{\mathbf{c} \in \mathbb{R}^n} \sum_{\mathbf{s}_i \in \mathfrak{D}_e} \sum_{(t, t') \in \hat{\pi}_i} \mathbf{w}_e(t) (\mathbf{s}_i(t') - \mathbf{c}(t))^2. \quad (5.8)$$

Или же для mwDTW:

$$\mathbf{c}_e = \operatorname{argmin}_{\mathbf{c} \in \mathbb{R}^n} \sum_{\mathbf{s}_i \in \mathfrak{D}_e} \sum_{(t, t') \in \hat{\pi}_i} \mathbf{W}_e(t, t') (\mathbf{s}_i(t') - \mathbf{c}(t))^2, \quad (5.9)$$

где $\hat{\pi}_i$ — векторно- (матрично-) взвешенный выравнивающий путь между временными рядами \mathbf{s}_i и \mathbf{c} .

Решение задачи нахождения центроида методом DBA. При замене единичного вектора весов в функции расстояния vwDTW на произвольный справедливо следующее

Следствие 2. При использовании произвольного вектора весов центроида \mathbf{w} (замене DTW на vwDTW с вектором весов \mathbf{w}) в задаче оптимизации (5.8) алгоритм DBA вычисления центроида находит локальный минимум при замене множества путей наименьшей стоимости $\{\tilde{\pi}\}_{i=1}^m$ на множество взвешенных путей наименьшей стоимости $\{\hat{\pi}_i\}_{i=1}^m$.

Доказательство данного следствия повторяет доказательство теоремы 2 при замене множества путей наименьшей стоимости $\{\tilde{\pi}\}_{i=1}^m$ на множество взвешенных путей наименьшей стоимости $\{\hat{\pi}_i\}_{i=1}^m$.

Для функции расстояния mwDTW следствие сохраняет свою формулировку и доказательство при замене вектора весов центроида на его матрицу весов \mathbf{W} .

Оптимизация и ограничения вектора весов. Положим теперь множество центроидов \mathbf{C} фиксированным. Каждому центроиду \mathbf{c}_e из множества \mathbf{C} поставлен в соответствие вектор неотрицательных весов \mathbf{w}_e , принадлежащий множеству $\hat{\mathbf{W}}$. Значения данного вектора весов выделяют наиболее типичные для класса участки центроида, сопоставляя им большие веса. Вычислим этот вектор, решая задачу оптимизации

$$\mathbf{w}_e = \operatorname{argmin}_{\mathbf{w} \in \mathbb{R}^n} \sum_{\mathbf{s}_i \in \mathfrak{D}_e} \sum_{(t, t') \in \pi_i} \mathbf{w}(t) (\mathbf{s}_i(t') - \mathbf{c}_e(t))^2. \quad (5.10)$$

При отсутствии ограничений на веса \mathbf{w}_e минимум (5.10) достигается при $\mathbf{w}_e = \mathbf{0}$. Для того чтобы избежать такого тривиального решения, введем ограничения на сумму элементов вектора весов

$$\sum_{t=1}^T \mathbf{w}_e(t) = T.$$

Предположим, что при решении задачи (5.10) нашлось t , для которого выполняется

$$\sum_{\mathbf{s}_i \in \mathfrak{D}_e} \sum_{t':(t,t') \in \pi_i} (\mathbf{s}_i(t') - \mathbf{c}_e(t))^2 = 0.$$

Для таких t элемент решения задачи оптимизации (5.10) $\mathbf{w}_e(t)$ примет большие значения, которые обеспечат выполнение ограничений на сумму элементов. Это приведет к локальному скоплению больших значений вектора весов, что сильно ухудшит дальнейшую интерпретацию вектора весов, а также сделает метод чувствительным к малым изменениям входных данных. Поэтому введем ограничения на элементы вектора весов сверху:

$$\mathbf{w}_e(t) \leq const, \quad t \in \{1, \dots, T\}.$$

Таким образом, исходная задача (5.10) примет следующий вид:

$$\begin{aligned} \mathbf{w}_e &= \operatorname{argmin}_{\mathbf{w} \in \mathbb{R}^n} \sum_{\mathbf{s}_i \in \mathfrak{D}_e} \sum_{(t,t') \in \pi_i} \mathbf{w}(t) (\mathbf{s}_i(t') - \mathbf{c}_e(t))^2, \\ &\sum_{t=1}^T \mathbf{w}_e(t) = T, \quad 0 \leq \mathbf{w}_e(t) \leq const, \quad t \in \{1, \dots, T\}, \end{aligned} \tag{5.11}$$

где $const$ — некоторая заданная константа.

Оптимизация матрицы весов. Ограничения матрицы весов. Как и при оптимизации вектора весов положим множество центроидов \mathbf{C} фиксированным. Каждому центроиду \mathbf{c}_e некоторого класса e из множества \mathbf{C} сопоставлена матрица неотрицательных весов \mathbf{W}_e , принадлежащая множеству $\hat{\mathbf{W}}$.

Используя те же соображения, что и для случая использования расстояния $vwDTW$, определим задачу нахождения матрицы весов центроида как задачу оптимизации с ограничениями при использовании расстояния $mwDTW$:

$$\begin{aligned} \mathbf{W}_e &= \operatorname{argmin}_{\mathbf{W} \in \mathbb{R}^{n \times n}} \sum_{\mathbf{s}_i \in \mathfrak{D}_e} \sum_{(t,t') \in \pi_i} \mathbf{W}(t, t') (\mathbf{s}_i(t') - \mathbf{c}_e(t))^2, \\ &\sum_{t=1}^T \sum_{t'=1}^T \mathbf{W}_e(t, t') = T^2, \quad 0 \leq \mathbf{W}_e(t) \leq const, \quad t \in \{1, \dots, T\}, \end{aligned} \tag{5.12}$$

где $const$ — некоторая заданная константа.

Сглаживание полученной матрицы. Полученная матрица не является устойчивой к изменению входных данных: при использовании других временных рядов выравнивающие пути будут иметь другой вид, а значит, и решение задачи оптимизации будет другое. Более устойчивой матрица получится после процедуры сглаживания.

Будем говорить, что элемент матрицы $\mathbf{W}_e(t, t')$ содержится в множестве Φ , если существует временной ряд $\mathbf{s}_i \in \mathfrak{D}_e$ такой, что путь наименьшей матрично-взвешенной стоимости проходит через элемент $\Omega(t, t')$ в матрице Ω , построенной для временного ряда \mathbf{s}_i и центроида.

При решении задачи оптимизации изменяться в меньшую сторону будут элементы $\{\mathbf{W}_e(t, t')\} \in \Phi$. При этом $\mathbf{W}_e(t, t') \notin \Phi$ достигнет своей верхней границы для выполнения ограничений, накладываемых на сумму элементов матрицы.

Выберем произвольный элемент матрицы весов $\mathbf{W}_e(t, t') \notin \Phi$. Вероятность того, что при добавлении нового временного ряда (например из тестовой выборки) выполнится $\mathbf{W}_e(t, t') \in \Phi$, выше, если среди ближайших к $\mathbf{W}_e(t, t')$ элементов в строке матрицы весов многие содержатся в множестве Φ . Тогда значение такого элемента должно быть похожим на значения соседних. Добьемся этого, выполнив сглаживание матрицы весов центроида:

$$\widetilde{\mathbf{W}}_e(t, t') = \frac{1}{2\varepsilon} \sum_{k=-\varepsilon}^{\varepsilon} \mathbf{W}_e(t, t' + k),$$

где ε — величина окна сглаживания, а $\widetilde{\mathbf{W}}_e$ — искомая матрица весов центроида.

Задача оптимизации параметров модели. Задача оптимизации параметров модели сведена к комбинации задач оптимизации (5.8) и (5.11), (5.12) для vwDTW:

$$\begin{aligned} \mathbf{w}_e, \mathbf{c}_e &= \underset{\mathbf{c}, \mathbf{w} \in \mathbb{R}^n}{\operatorname{argmin}} \sum_{\mathbf{s}_i \in \mathfrak{D}_e} \sum_{(t, t') \in \pi_i} (\mathbf{w}(t)(\mathbf{s}_i(t') - \mathbf{c}(t))^2), \quad e = 1, \dots, E, \\ &\sum_{t=1}^T \mathbf{w}_e(t) = T, \quad 0 \leq \mathbf{w}_e(t) \leq const, \quad t \in \{1, \dots, T\}. \end{aligned} \tag{5.13}$$

Или же для mwDTW:

$$\begin{aligned} \mathbf{W}_e, \mathbf{c}_e &= \underset{\mathbf{W} \in \mathbb{R}^{n \times \mathbb{R}^n}}{\operatorname{argmin}} \sum_{\mathbf{s}_i \in \mathfrak{D}_e} \sum_{(t, t') \in \pi_i} (\mathbf{W}(t, t')(\mathbf{s}_i(t') - \mathbf{c}_e(t))^2), \quad e = 1, \dots, E, \\ &\sum_{t=1}^T \sum_{t'=1}^T \mathbf{W}_e(t, t') = T^2, \quad 0 \leq \mathbf{W}_e(t) \leq const, \quad t \in \{1, \dots, T\}. \end{aligned} \tag{5.14}$$

Эту задачу будем решать, вычисляя сначала множество центроидов \mathbf{C} при фиксированном начальном приближении множества весов центроидов $\hat{\mathbf{W}}$, а затем

вычисляя множество весов центроидов $\hat{\mathbf{W}}$ при фиксированном множестве центроидов \mathbf{C} . Таким образом, алгоритм вычисления параметров модели будет иметь следующий вид:

Шаг 1. Начальное приближение вектора весов центроида:

$$\mathbf{w}_e = \mathbf{1}, \quad e = 1, \dots, E.$$

Или же начальное приближение матрицы весов центроида:

$$\mathbf{W}_e = \mathbf{1}, \quad e = 1, \dots, E.$$

Шаг 2. Начальное приближение центроида класса — произвольный элемент класса:

$$\mathbf{c}_e = \mathbf{s}_j \in \mathcal{D}_e, \quad e = 1, \dots, E.$$

Шаг 3. Вычисление центроида при фиксированном векторе (матрице) весов центроида как решение задачи оптимизации (5.8).

Шаг 4. Вычисление вектора (матрицы) весов центроида при фиксированном центроиде как решение задачи оптимизации (5.11), (5.12).

5.2. Свойства множества отклонений выравнивающего пути

Алгоритм DTW выравнивает два временных ряда и строит множество отклонений, усредняя его. При этом теряется информация о распределении множества отклонений между двумя временными рядами. Предлагается определить вектор-функцию расстояния между временными рядами, которая позволит сохранить эту информацию и дать более высокое качество при попарном анализе принадлежности объектов к одному классу эквивалентности.

Вектор-функция описывает распределение элементов множества отклонений. Признаковое описание объекта в задаче классификации — это набор значений некоторых статистик от множества отклонений. Они вычисляются между данным объектом и всеми эталонными объектами. Дополнительная информация о распределении элементов множества отклонений улучшает качество классификации. В статье автора [6] предлагается метод классификации и демонстрируются результаты его работы в задаче классификации физической активности человека по данным акселерометра мобильного телефона.

5.2.1. Постановка задачи поиска оптимального классификатора и функции расстояния

Предположим, что решается задача метрической классификации объектов. Напомним, что вектор-функция расстояния между двумя временными рядами \mathbf{s}_1 и \mathbf{s}_2 — это отображение ρ из пространства пар временных рядов в пространство вещественных чисел размерности l :

$$\rho : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^l.$$

Присвоим каждому объекту \mathbf{s}_i признаковое описание \mathbf{x}_i . Оно состоит из расстояний между данным временным рядом \mathbf{s}_i и всеми эталонными объектами \mathbf{c}_e :

$$\mathbf{x}_i = \begin{bmatrix} \rho(\mathbf{s}_i, \mathbf{c}_1) & \vdots & \dots & \vdots & \rho(\mathbf{s}_i, \mathbf{c}_{|\mathbb{Y}|}) \end{bmatrix}^T, \quad \mathbf{x}_i \in \mathbb{R}^{l|\mathbb{Y}|}.$$

Здесь $\mathbf{a}:\mathbf{b}$ обозначает конкатенацию векторов.

Модель классификации обучается по полученным признакам объектов $\mathfrak{X} = \{(\mathbf{x}_i, y_i)\}, i \in \{1, \dots, N\}, \mathbf{x}_i \in \mathbb{R}^{l|\mathbb{Y}|}$.

Таким образом, модель классификации f является отображением из пространства признаков объекта $\mathbf{x} \in \mathbb{R}^{l|\mathbb{Y}|}$ и параметров модели $\mathbf{w} \in \mathbb{W}$ в множество меток классов:

$$f : \mathbb{R}^{l|\mathbb{Y}|} \times \mathbb{W} \rightarrow \mathbb{Y}.$$

Будем предполагать, что f — модель из множества $\mathfrak{F} = \{\text{Random Forest, KNN, SVM}\}$. Параметры модели \mathbf{w} оптимизируют функцию качества Q , и оптимальные параметры обозначаются \mathbf{w}^* :

$$\mathbf{w}^* = \arg \min_{\mathbf{w} \in \mathbb{W}} Q(\mathbf{w} | \mathfrak{X}, f). \quad (5.15)$$

Используя (5.15) с фиксированными f и Q , мы получаем различные результаты классификации для разных функций расстояния ρ . Обозначим $\mathfrak{R} \ni \rho$ — множество функций расстояния, такое, что \mathfrak{R} включает в себя модификации метода динамического выравнивания временных рядов: DTW, DTW с Гауссовским ядром (2.2) и другие.

Обозначим \mathfrak{X}_ρ признаковое описание объекта, полученное с помощью фиксированной функции расстояния ρ . Необходимо найти функцию расстояния ρ и модель классификации f , такие, что они оптимизируют функцию качества Q :

$$(\rho^*, f^*) = \arg \min_{(\rho, f) \in \mathfrak{R} \times \mathfrak{F}} Q(\mathbf{w}_\rho^* | \mathfrak{X}_\rho, f), \quad (5.16)$$

где оптимальные параметры \mathbf{w}_ρ^* получены из (5.15) для фиксированных ρ и f .

5.2.2. Анализ отклонений

Пусть $\widehat{\pi}$ — выравнивающий путь между временным рядом \mathbf{s} и эталонным объектом \mathbf{c} . Напомним, что $\{\Omega_{ij}\}, (i, j) \in \widehat{\pi}$ называется множеством отклонений и обозначается δ : $\delta = \{\Omega_{ij}\}, (i, j) \in \widehat{\pi}$.

Пусть отображение $\mathbf{d} : (\mathbf{s}, \mathbf{c}, \mathbf{a}(\mathbf{s}, \mathbf{c})) \mapsto \delta$ сопоставляет паре временных рядов \mathbf{s}, \mathbf{c} и выравнивающему пути $\widehat{\pi}$ между ними множество отклонений δ :

$$\mathbf{d} : \mathbb{R}^n \times \mathbb{R}^n \times (\mathcal{I}_s \times \mathcal{I}_c)^{|\widehat{\pi}|} \rightarrow \mathbb{R}^{|\widehat{\pi}|}.$$

Вариационный ряд элементов множества δ описывает различия между парой временных рядов. Функция расстояния ρ , основанная на стандартном алгоритме динамического выравнивания, сопоставляет множеству δ одно число путем усреднения по δ :

$$\rho(\mathbf{s}, \mathbf{c}) = \frac{\sum_{\delta_i \in \delta} \delta_i}{\sum_{\delta_i \in \delta} 1}. \quad (5.17)$$

Предлагается рассмотреть статистики, отличные от среднего значения, для того чтобы описать распределение элементов δ : α -квантили множества δ и средние значения на таких его подмножествах, элементы которых больше, чем заранее заданный α -квантиль. Качество классификации (5.16) улучшается в новом признаковом пространстве. Обозначим $q_\alpha(\delta)$ — α -квантили и $p_\alpha(\delta)$ — средние по соответствующему подмножеству:

$$p_\alpha(\delta) = \frac{\sum_{r \in \delta} r[r > q_\alpha(\delta)]}{\sum_{r \in \delta} [r > q_\alpha(\delta)]}.$$

Пусть итоговая вектор-функция расстояния ρ возвращает конкатенацию $q_\alpha(\delta)$ и $p_\alpha(\delta)$ для некоторых α .

Определение 31. Обозначим через δDTW функцию расстояния между временными рядами \mathbf{s} и \mathbf{c} , для которой зафиксирован алгоритм построения пути \mathbf{a} , отображение \mathbf{d} и набор чисел $\alpha = \{\alpha_i\}$, $i \in \{1, \dots, a\}$:

$$\begin{aligned} \rho(\mathbf{s}, \mathbf{c} | \mathbf{a}, \mathbf{d}) &= [q_{\alpha_1}(\delta), \dots, q_{\alpha_a}(\delta), p_{\alpha_1}(\delta), \dots, p_{\alpha_a}(\delta)]^\top, \\ \delta &= \mathbf{d}(\mathbf{s}, \mathbf{c}, \mathbf{a}(\mathbf{s}, \mathbf{c})). \end{aligned} \quad (5.18)$$

5.3. Выравнивание скрытых представлений (TBD)

К определению пространственно-временных объектов могут быть сведены объекты произвольной природы, которые изменяются во времени. Даже текст, который был написан человеком, является примером такого объекта.

Предлагается использовать понятие эмбеддинга для обобщения метода выравнивания на объект произвольной природы. Представим, что у нас существует модель векторного представления для произвольного объекта. Объект изменяется во времени, и векторное представление изменяется вместе с ним.

В таком случае задача может быть сведена к многомерному выравниванию, так как набор векторных представлений во времени является многомерным временным рядом.

Примем два пространственно-временных объекта для выравнивания как $\mathbf{S}_1(t)$, $\mathbf{S}_2(t)$.

Также определим функцию, создающую скрытое представление, через $g(\mathbf{S})$:

Определение 32. Скрытым представлением объекта \mathbf{S} назовем вектор $\hat{\mathbf{s}}$, который обладает размерностью, много меньшей размерности исходного объекта.

При помощи функции g мы можем перевести исходные объекты $\mathbf{S}_1(t)$, $\mathbf{S}_2(t)$, между которыми может быть трудозатратно посчитать расстояние, в объекты, вектора $\hat{\mathbf{s}}_1(t)$, $\hat{\mathbf{s}}_2(t)$.

К описанным объектам, многомерным временным рядам, можно применить подход, описанный в главе с алгоритмом анализа многомерных временных рядов.

Глава 6

Анализ методов метрической классификации путем постановки вычислительных экспериментов

6.1. Вычислительный эксперимент по метрической классификации методом ближайшего соседа (раздел 2.2)

Прикладное использование функции расстояния. Задана выборка $\mathfrak{D} = \{(\mathbf{s}_i, y_i)\}_{i=1}^m$, состоящая из пар объект — ответ. Объектами служат объекты сложной структуры: $\mathbf{s}_i \in \mathbb{R}^{n \times n}$, а ответами являются метки класса — $y_i \in \mathbb{Y} = \{1, \dots, E\}$, где $E \ll m$. Выборка разделена на обучение $\mathfrak{D}_l = \{(\mathbf{s}_i, y_i)\}_{i=1}^{m_1}$ и контроль $\mathfrak{D}_t = \{(\mathbf{s}_i, y_i)\}_{i=m_1+1}^{m_1+m_2}$.

Пусть модель классификации f принадлежит множеству моделей метрической классификации 1NN, которые классифицируемому объекту ставят в соответствие метку класса ближайшего объекта из обучающей выборки по заданной функции расстояния ρ :

$$\hat{y} = f(\mathbf{s}'|\rho) = y \underset{i=1, \dots, m_1}{\operatorname{argmin}} \rho(s', s_i)$$

. Критерий качества Q модели f для задачи классификации — доля правильно проставленного класса на контрольной выборке:

$$Q(f|\rho) = \frac{1}{m_2} \sum_{i=m_1}^{m_1+m_2} [f(\mathbf{s}_i|\rho) = y_i].$$

Требуется выбрать функцию расстояния ρ для модели классификации $f : \mathbb{R}^{n \times n} \rightarrow \mathbb{Y}$, максимизирующую критерий качества Q на контрольной выборке:

$$\rho = \underset{\rho \in \{\text{mDTW}, L_2\}}{\operatorname{argmax}} (Q(f|\rho)). \quad (6.1)$$

Вычислительный эксперимент проведен на модельных данных с допустимыми преобразованиями, и на реальных данных: объектах коллекции MNIST, с допустимыми преобразованиями, на спектrogramмах зашумленных сигналов.

Решается задача метрической классификации методом ближайшего соседа. В таблице 6.1 приведены значения критерия качества функции расстояния $Q_\rho(\mathfrak{D})$ и критерия качества метрической классификации $Q(f|\rho)$ при использовании двух функций расстояния: предложенной в работе mDTW и L_2 .

Модельные данные — это нулевые матрицы со случайными ненулевыми строками, столбцами, подпрямоугольниками с наложенным шумом. К ним применены допустимые преобразования, согласованные с гипотезой наличия локальных и глобальных искажений. На Рис. 6.1 показан пример оптимального выравнивания двух объектов. Линиями показаны элементы пути $\hat{\pi}$.

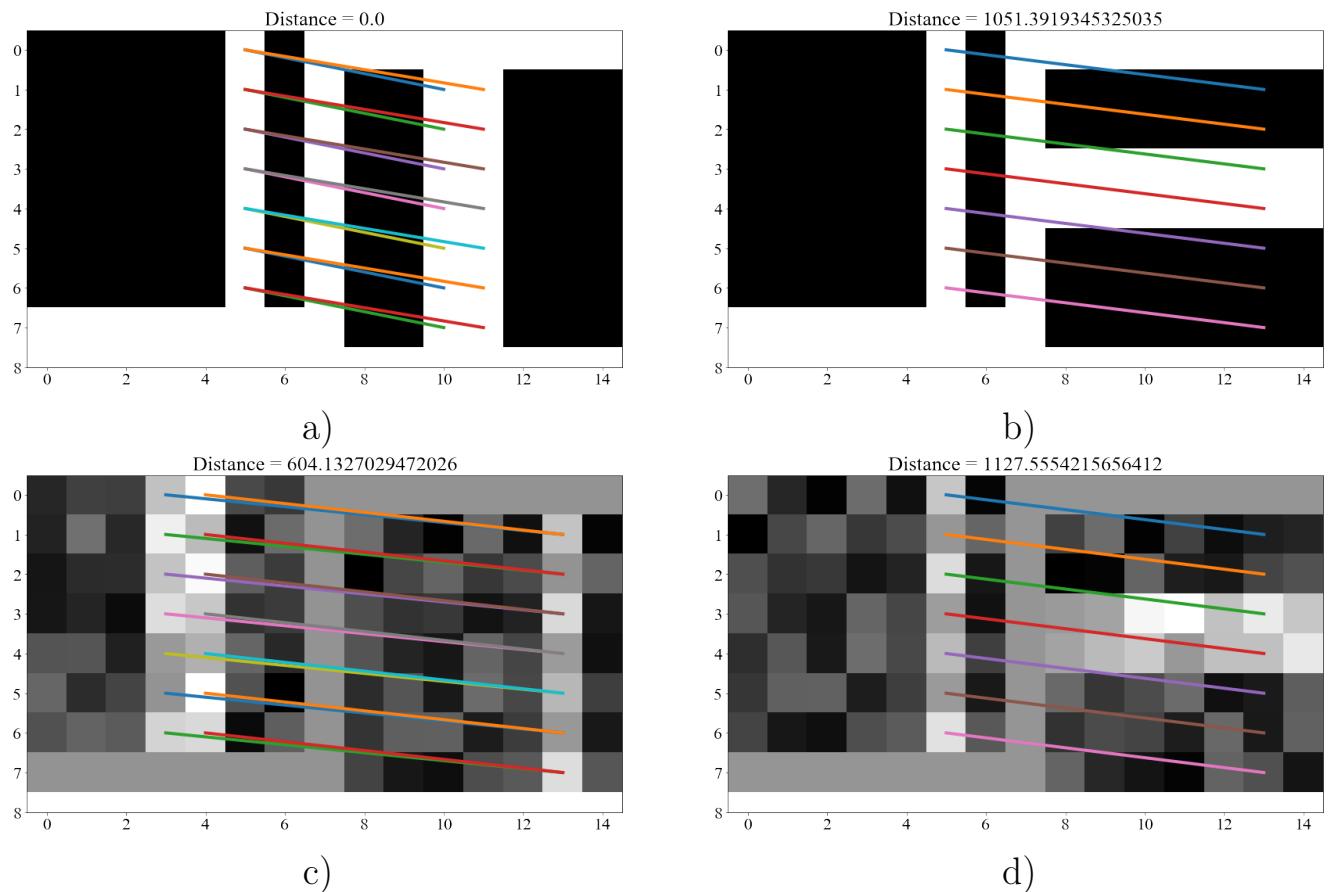


Рис. 6.1. Выравнивание модельных данных: а) один класс без шума, б) разные классы без шума, в) один класс с шумом, г) разные классы с шумом.

Подготовлена подвыборка набора данных MNIST. Она состоит из 100 объектов классов: 0 и 1 сниженной размерности с допустимыми преобразованиями. На Рис. 6.2 показан пример оптимального выравнивания объектов.

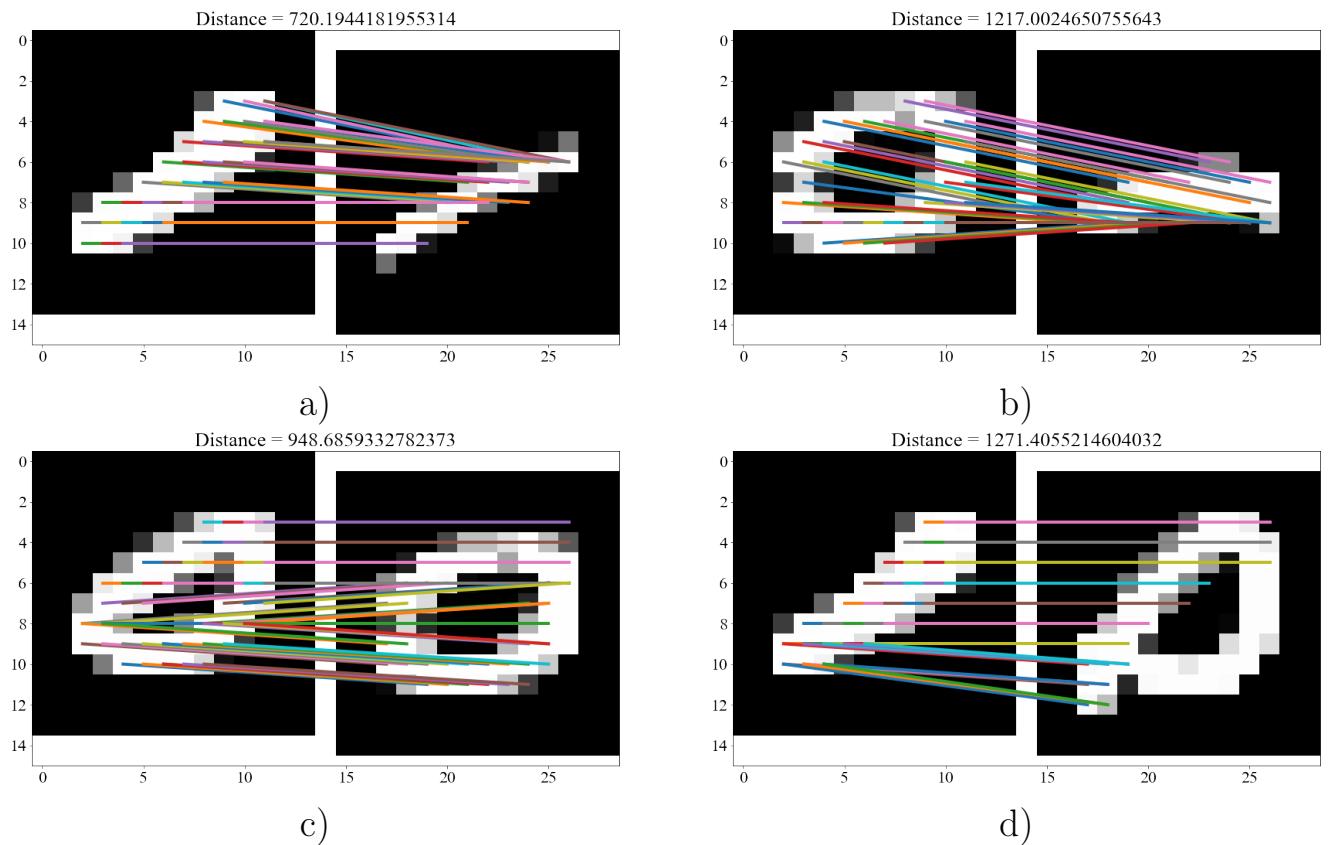


Рис. 6.2. Выравнивание данных MNIST: а) один класс, б) разные классы, в) один класс, г) разные классы.

Аналогичный эксперимент проведен для решения задачи метрической классификации спектров различных сигналов, пример которых приведен на Рис. 6.3. На рисунке показаны примеры Фурье-спектров этих сигналов. Спектр получен путем применения быстрого преобразования Фурье к исходному сигналу для различных окон с фиксированным размером и сдвигом. Исходные временные ряды обладали свойством периодичности, период выбирался случайным образом.

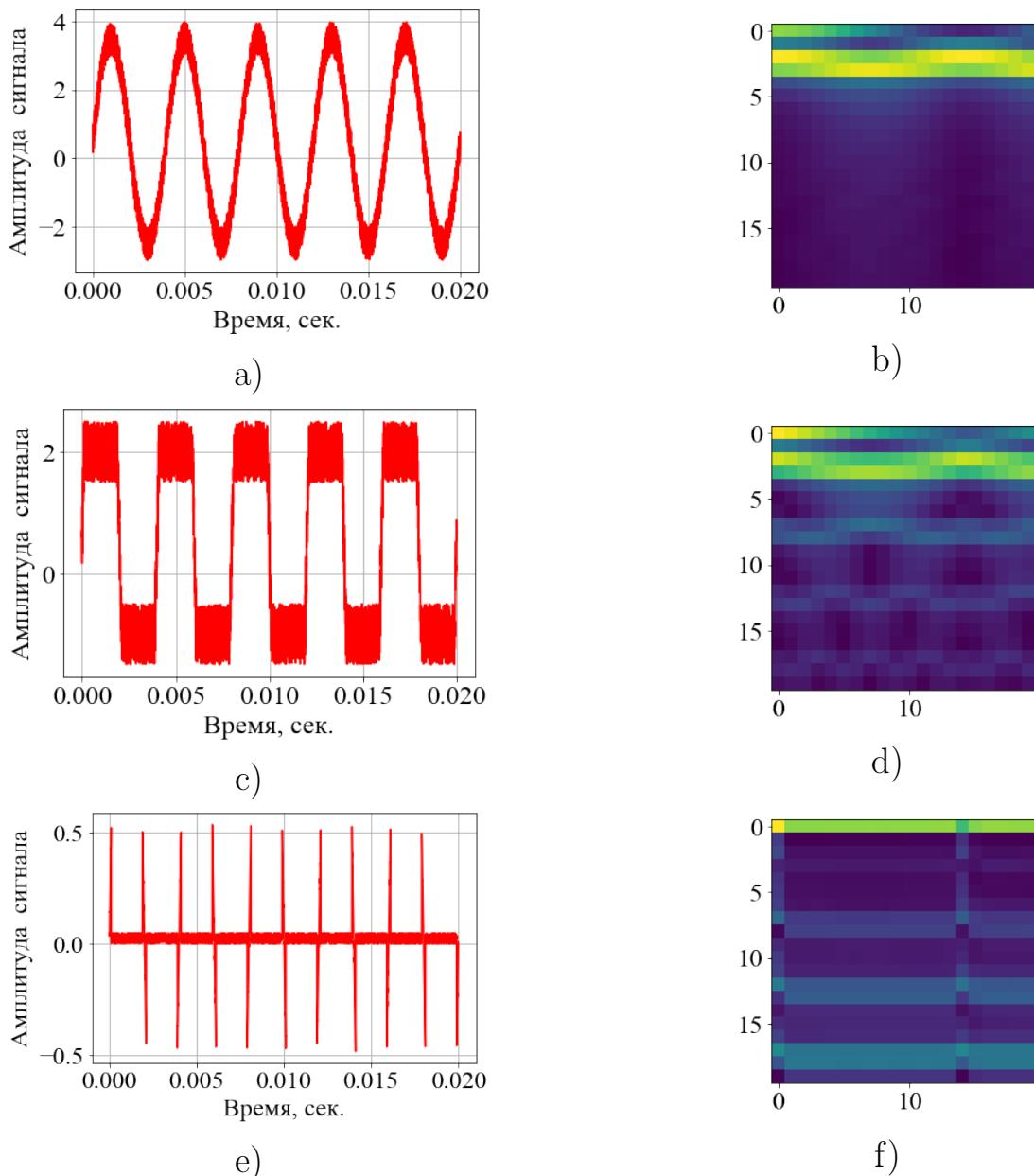


Рис. 6.3. Данные спектров сигналов: а) класс 1, б) спектр класса 1, в) класс 2, г) спектр класса 2, д) класс 3, е) спектр класса 3.

Тестирование проведено на различного рода данных: исходных модельных данных без наложения допустимых преобразований, с ними, а также на модельных данных с наложенным поверх объектов случайным шумом.

В каждом из проведенных экспериментов была продемонстрирована устойчивость предложенного подхода к допустимым преобразованиям. Наилучшее значение критерия качества задачи классификации было достигнуто при использовании предложенной функции расстояния.

Таблица 6.1. Снижение расстояний при выполнении преобразований для различных наборов данных.

| Данные / Метод | L_2 | | MatrixDTW | |
|---|-------------|------------------------|-------------|------------------------|
| | $Q(f \rho)$ | $Q_\rho(\mathfrak{D})$ | $Q(f \rho)$ | $Q_\rho(\mathfrak{D})$ |
| Модельные данные без преобразований | 92% | 78% | 100% | 85% |
| Модельные данные с преобразованиями | 86% | 65% | 100% | 82% |
| Модельные данные с преобразованиями и шумом | 69% | 61% | 92% | 78% |
| MNIST без преобразований | 95% | - | 95% | - |
| MNIST с преобразованиями | 53% | - | 92% | - |
| Спектр сигнала | 83% | - | 96% | - |

6.2. Вычислительный эксперимент по классификации синтетических и реальных временных рядов (раздел 2.4)

Для проверки свойств введенной функции расстояния, а также выбранной модели был проведен вычислительный эксперимент на реальных и синтетических данных. Свойства функций расстояния vwDTW и mwDTW для наглядности продемонстрированы на синтетических данных, представляющих собой смещенные и линейно деформированные временные ряды аналитических функций: $\sin x$, \sqrt{x} , x^2 — длиной 100 точек. В выборке находилось 100 временных рядов каждого класса: 50 в обучающей и 50 в контрольной. Обозначим классы как 1 — $\sin x$, 2 — \sqrt{x} , 3 — x^2 соответственно.

Пример такой выборки и результаты нахождения параметров модели по обучающей выборке показаны на Рис. 6.4. По оси абсцисс отложены значения времени, а по оси ординат — значения временного ряда. На левом верхнем графике приведены примеры аналитических функций, используемых в создании синтетической выборки временных рядов. На правом верхнем графике изображены центроид (нижний временной ряд) и вектор весов (верхний временной ряд) для класса 1. Аналогично на нижних графиках показаны результаты для классов 2 и 3.

Векторы весов описывают наиболее информативные участки центроида. Так, для центроида класса 1 ($\sin x$) наиболее информативными оказались минимумы и максимумы, в отличие от точек перегиба.

Для сравнения свойств полученной функции расстояния с функцией расстояния DTW были посчитаны расстояния до центроидов для всех временных рядов контрольной выборки с помощью DTW и vwDTW, после чего производилась классификация. Каждому временному ряду контрольной выборки ставилась в соответствие метка того класса, расстояние до центроида которого было минимальным. Результат классификации с помощью функции расстояния vwDTW — 97%, а для функции расстояния DTW — 84%, что на 15% меньше. Построенная в работе функция расстояния лучше разделила временные ряды разных классов, сгруппировала их вокруг соответствующих центроидов.

Также были построены матрицы весов центроида на синтетических данных. Вследствие хорошей интерпретируемости, в работе приведено визуальное отображение матрицы весов центроида только для класса 1. На Рис. 6.5 сверху слева изображены примеры временных рядов. Сверху справа — множество выравнивающих путей по матрице Ω . Нижний левый рисунок — матрица весов центроида до сглаживания. Правый левый рисунок — после сглаживания. Синие цвета соответствуют маленьким значениям элементов матрицы, а красные — большим. Хорошо просматривается периодичность в матрице, напоминающая периодичность вектора весов. Предполагается, что функция mwDTW будет лучше разделять классы, которые сильно различаются между собой, так как матрица весов центроида учитывает и среднее отклонение выравнивающего пути от диагонали в матрице Ω . При отклонениях пути сильнее типичного для

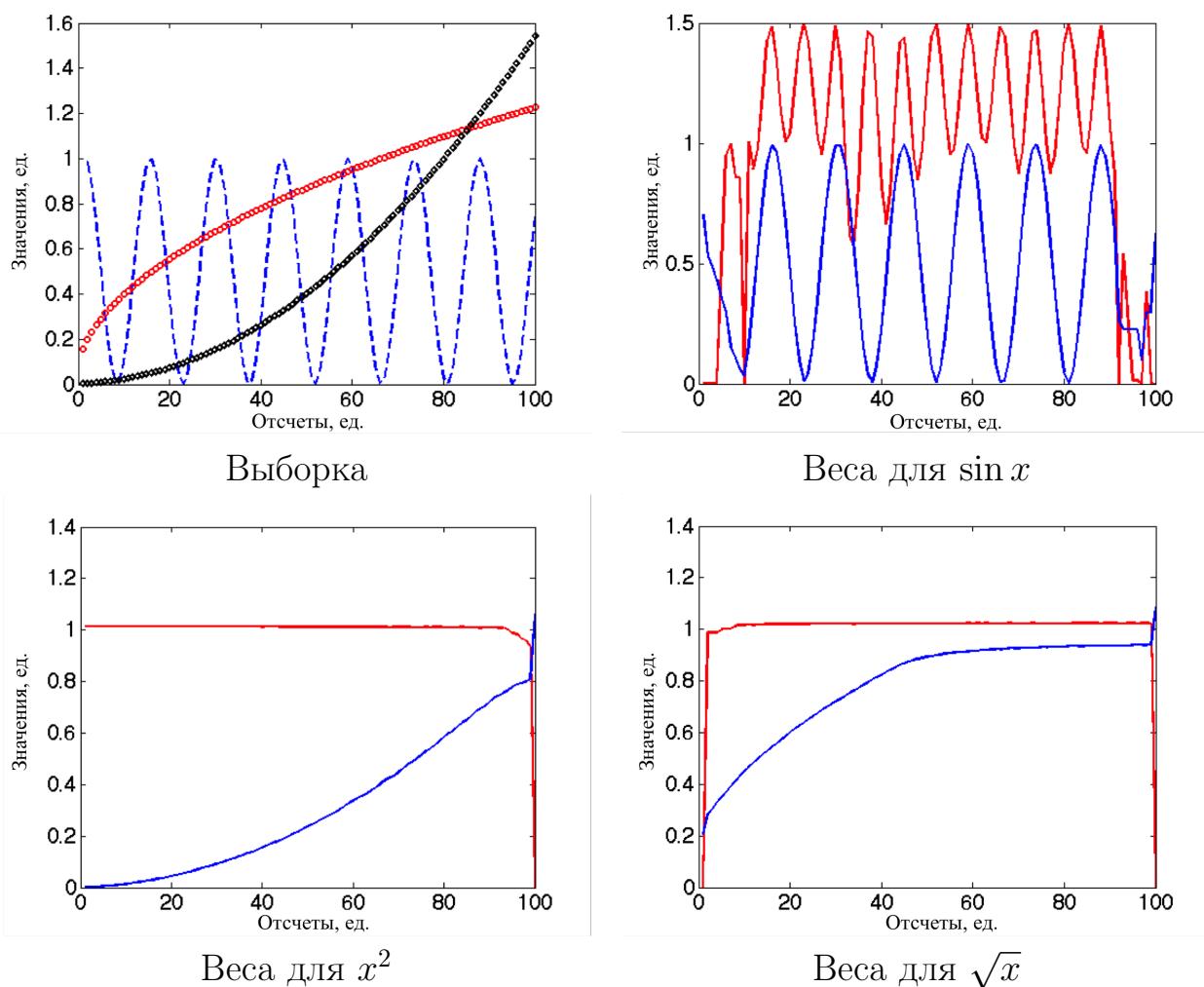


Рис. 6.4. Примеры синтетических временных рядов аналитических функций (выборка), результаты построения центроида и вектора весов для синтетических данных

данного класса, элементам пути будут приписываться большие веса, что видно из структуры матрицы весов центроида.

Использование функции `mwDTW` для классификации синтетических временных рядов улучшает классификацию по сравнению с `DTW`. Этот результат аналогичен случаю использования `vwDTW` — 97%.

Для демонстрации работы предложенной модели на реальных данных и ее сравнения с базовой моделью классификации были использованы данные акселерометра мобильного телефона. Вследствие большой вычислительной сложности они сравнивались на данных, представляющих собой 600 временных рядов длиной 200 точек, каждый из которых представляет собой абсолютные значения ускорения мобильного телефона, объединяя три временных ряда: временной ряд ускорения по оси X (200 измерений), оси Y (200 измерений) и оси Z (200 измерений). Выделено шесть типов физической активности: ходьба, бег, сидение, стояние, подъем, спуск. Временные ряды записывались акселерометром, который находился в кармане у человека, выполняющего один из типов физи-

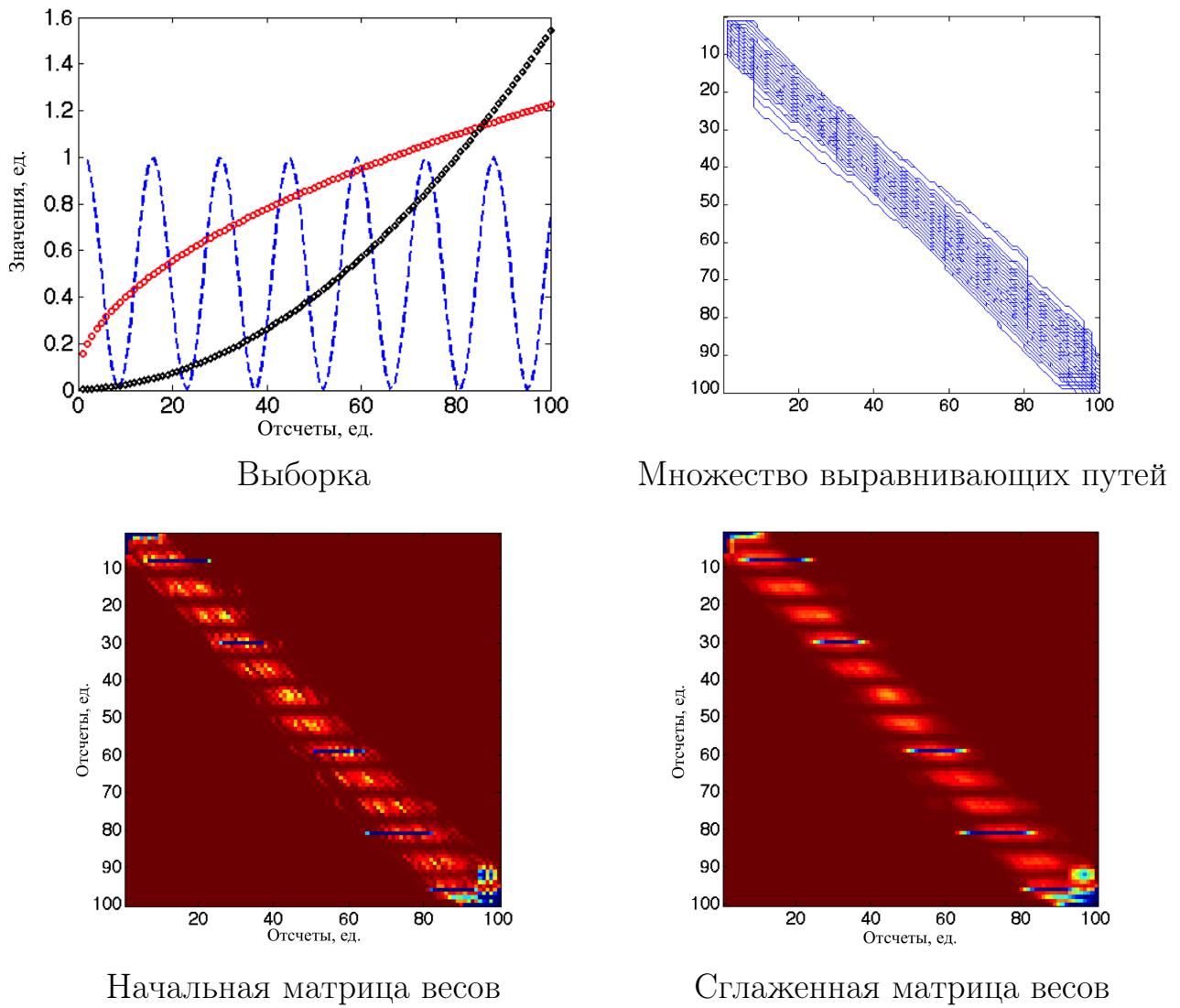


Рис. 6.5. Примеры синтетических временных рядов аналитических функций (выборка), результат построения матрицы весов центроида класса $\sin(x)$

ческой активности, после чего разделялись на 10-секундные сегменты. Примеры таких временных рядов приведены на Рис. 6.6.

Данные разделялись на обучающую и контрольную выборку. В обучающую выборку входило по 70 временных рядов каждого вида физической активности, а в контрольную — по 30 временных рядов. Производилась классификации методом k ближайших соседей, использующим построенную матрицу попарных расстояний. Осуществлялся контроль качества при помощи кросс-валидации. В табл. 6.2 приведены результаты классификации при использовании новой модели, использующей функцию vwDTW, и базовой модели классификации.

Качество классификации базовой модели ухудшилось в сравнении с вычислительным экспериментом в работе [1], так как теперь используются абсолютные значения ускорения, а не последовательно соединенные временные ряды ускорения вдоль трех координат. Использование функции расстояния vwDTW улучшает классификацию для всех классов физической активности, повышая

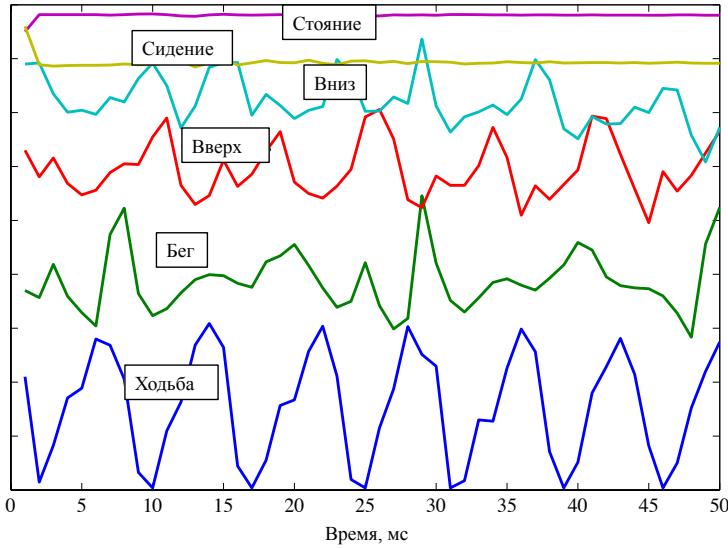


Рис. 6.6. Примеры временных рядов измерений акселерометра для разных видов физической активности

общий уровень классификации на 9%.

Таблица 6.2. Сравнение эффективности предложенной (vwDTW) и базовой модели классификации на данных

| Модель | Бег | Ходьба | Вверх | Вниз | Сидение | Стояние | Общее |
|--|-----|--------|-------|------|---------|---------|-------|
| vwDTW | | | | | | | |
| Точность по критерию скользящего контроля, % | 97 | 95 | 79 | 75 | 95 | 95 | 89 |
| DTW [1] | | | | | | | |
| Точность по критерию скользящего контроля, % | 95 | 92 | 60 | 60 | 85 | 90 | 80 |

Полученные параметры модели для реальных данных изображены на Рис. 6.7. Для класса бег, например, хорошо просматривается периодичность временного ряда как для центроида, так и для вектора весов центроида.

Для сравнения модели, использующей mwDTW, с базовой моделью классификации были использованы данные, представляющие собой 600 временных рядов длиной 100 точек. Временные ряды такой длины выбраны для разумного ограничения времени работы методов численной оптимизации. В остальном этот эксперимент повторяет тот, что выполнен для сравнения базовой модели и модели, использующей vwDTW. Результаты классификации приведены в табл. 6.3. Видно, что длина временных рядов, а следовательно, и количество периодов временного ряда, существенно влияют на качество классификации.

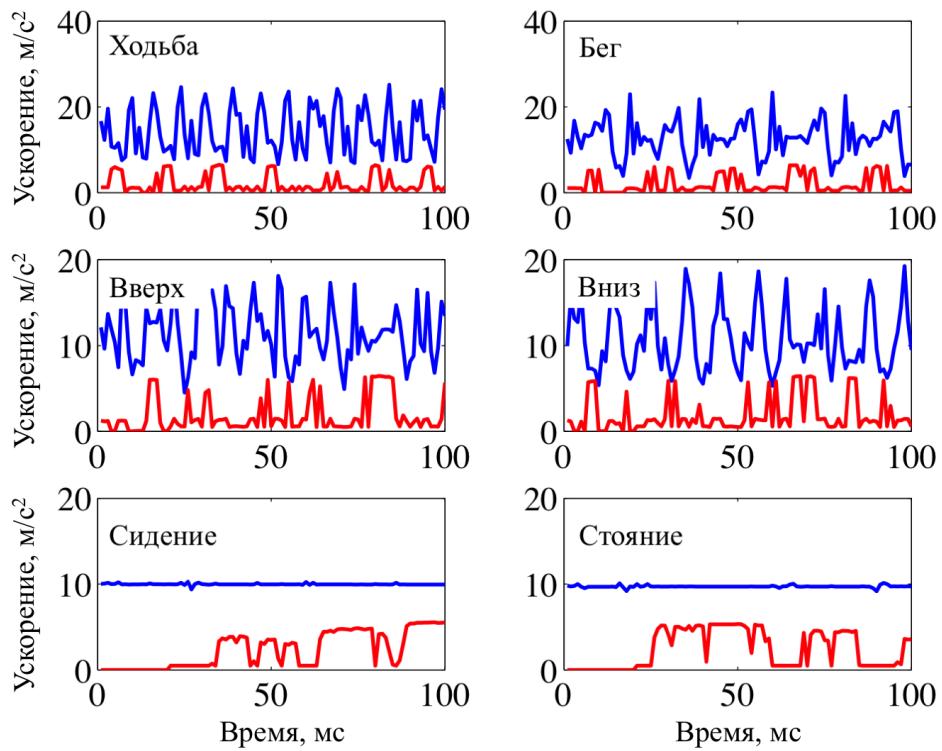


Рис. 6.7. Примеры временных рядов измерений акселерометра для разных видов физической активности

Качество базовой модели сильно упало по сравнению с временными рядами длиной 200 точек. При этом использование mwDTW улучшает классификацию на 13%.

Таблица 6.3. Сравнение эффективности предложенной (mwDTW) и базовой модели классификации и алгоритма разделяющей классификации на данных

| Модель | Бег | Ходьба | Вверх | Вниз | Сидение | Стояние | Общее |
|--|-----|--------|-------|------|---------|---------|-------|
| mwDTW | | | | | | | |
| Точность по критерию скользящего контроля, % | 95 | 95 | 78 | 76 | 90 | 90 | 87 |
| DTW | | | | | | | |
| Точность по критерию скользящего контроля, % | 88 | 83 | 55 | 60 | 82 | 80 | 74 |

6.3. Вычислительный эксперимент по классификации временных рядов методом ближайшего центроида (раздел 2.6)

Исследуются свойства функции расстояния, предложенной в разделе 2.6. Используемые данные представляют собой набор временных рядов, описывающих активность человека [55].

Вычислительный эксперимент проводится следующим образом. Выборка делится на шесть подмножеств, соответствующих типам активности человека и для каждого подмножества находятся центроиды. По выборке и центроидам строятся соответствующие им непрерывные объекты. Исследуются зависимости расстояния DTW и времени вычислений от числа узлов сплайна и находится наилучшее число узлов.

Исходные данные — дискретные временные ряды, каждый из которых является последовательностью измерений $\{s_i\}_{i=1}^n$. Для дальнейшей интерполяции представим их в виде $\{(x_i, y_i)\}_{i=1}^n$, где $x_i = i$, $y_i = s_i$, $i = 1, \dots, n$.

Для интерполяции используются кубические сплайны. Пример такой интерполяции приведен на Рис. 6.8. На рисунке точки обозначают дискретный временной ряд, а непрерывная линия — его кубическая интерполяция. Вообще говоря, для получения непрерывных объектов может быть использован любой другой способ интерполяции.

Интерполяция кубическими сплайнами была выбрана, потому что она удобна для вычисления интеграла из определения стоимости пути.

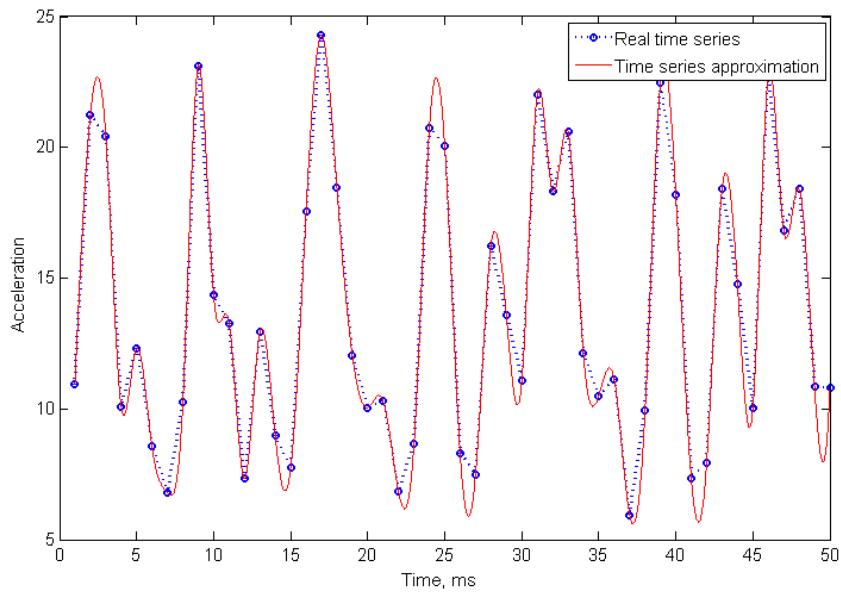


Рис. 6.8. Интерполяция кубическими сплайнами

То же самое семейство параметрических функций используется для аппроксимации выравнивающего пути. Число узлов N является гиперпараметром, а их координаты вдоль первой (временной) оси фиксированы. Их координаты

вдоль второй оси — параметры модели. Если N мало, то наша аппроксимация недостаточно точно приблизит истинный выравнивающий путь и его стоимость будет завышена. С другой стороны, вычислительная сложность растет при увеличении N . На Рис. 6.9 показан результат экспериментальной проверки этих утверждений. Расстояние и вычислительная сложность были измерены для каждого из классов по отдельности, а затем результаты усреднялись. На Рис. 6.9 изображены средние и стандартные отклонения этих измерений. Количество узлов было выбрано в соответствии с этими результатами.

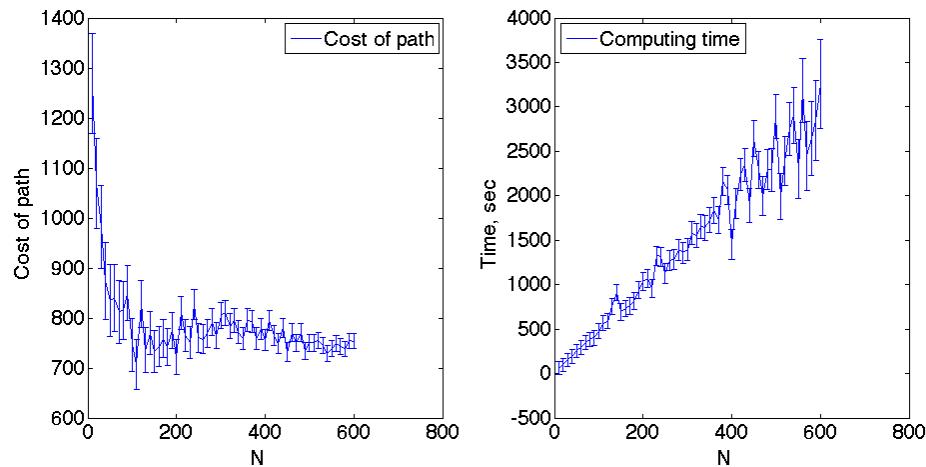


Рис. 6.9. Зависимость стоимости выравнивающего пути и времени вычислений от числа узлов

Для исследования свойств расстояния мы строим матрицу расстояний между всеми временными рядами и центроидами и усредняем ее для каждого класса по отдельности. Средние значения расстояний для каждого класса приведены в таблице 6.4. В каждой строке записаны средние расстояния между одним классом и всеми центроидами. Для каждого класса ближайший центронд соответствует этому же классу.

Таблица 6.4. Средние значения для всех классов

| | Ходьба | Бег | Подъем | Спуск | Сидение | Лежание |
|---------|------------|------------|------------|------------|----------|-----------|
| Ходьба | 693 | 803 | 811 | 733 | 1165 | 1143 |
| Бег | 676 | 498 | 696 | 610 | 946 | 927 |
| Подъем | 714 | 739 | 696 | 701 | 1038 | 1021 |
| Спуск | 591 | 601 | 653 | 464 | 836 | 804 |
| Сидение | 516 | 465 | 434 | 400 | 6 | 42 |
| Лежание | 508 | 441 | 454 | 366 | 105 | 79 |

Для решения задачи классификации использовался метод ближайшего центроида. Измерение точности классификации проводилось следующим образом:

один объект удалялся из выборки, центроиды строились для всех классов без использования этого объекта. Затем находились расстояния от удаленного временного ряда до всех центроидов, после чего рассматриваемому объекту присваивался класс ближайшего центроида.

Дискретный и непрерывный DTW дали следующие результаты: 85% и 83% точности соответственно. Таким образом, результаты отличаются не сильно. Объем хранимых данных уменьшился вдвое, а время вычислений увеличилось по сравнению с эффективной реализацией обычного DTW. Некоторые стандартные для рассматриваемой задачи подходы дают более хорошие результаты, чем предложенный подход (90% для нейронных сетей и 89% при использовании параметров аппроксимации для дальнейшей классификации). Но количество полученных экспериментальных результатов недостаточно для их статистической значимости.

6.4. Вычислительный эксперимент по многоклассовой классификации временных рядов методами Random forest, SVM, KNN (раздел 2.7)

Рассматривается задача метрической многоклассовой классификации. Использовались данные открытого датасета WISDM. Временные ряды из этого датасета описывают физическую активность человека. Данные собирались с помощью акселерометра мобильного телефона. В эксперименте исследуется предложенная функция расстояния δDTW и сравнивается качество классификации для различных функций расстояния $\mathfrak{R} = \{\text{DTW}, \text{Kernel DTW}, \delta\text{DTW}\}$.

Есть шесть типов активности человека: ходьба, бег, прыжки, стояние на месте, подъем и спуск по лестнице. Временные ряды имеют длину 6 с и шаг по времени 10^{-2} с. Каждый класс $e \in \mathbb{Y}$ содержит 650 временных рядов.

Выборка $\mathfrak{D} = \mathfrak{D}_l \cup \mathfrak{D}_t$ разделена на обучающую \mathfrak{D}_l и тестовую \mathfrak{D}_t .

Пусть \mathbf{c}_e — центроиды, построенные для всех классов e по обучающей выборке. Эти центроиды показаны на Рис. 6.10. Признаковые описания объектов \mathfrak{X}_ρ построены для фиксированной функции расстояния $\rho \in \mathfrak{R}$ для обучающей и тестовой выборки.

Параметры модели классификации \mathbf{w} оптимизируются (5.15) с использованием обучающей выборки. Функция качества Q — точность классификации. Модель и функция расстояния минимизируют ее (5.16) на тестовой выборке.

$$Q(\mathbf{w}_\rho | \mathfrak{X}_\rho, f) = \frac{\sum_i [y_i = f(\mathbf{x}_i, \mathbf{w}_\rho)] [s_i \in \mathfrak{D}_t]}{\sum_i [s_i \in \mathfrak{D}_t]}.$$

В эксперименте сравниваются результаты для четырех различных алгоритмов **a** выравнивания между двумя временными рядами.

- 1 “Евклидово ядро”. Этот алгоритм использует евклидово ядро (2.1) и работает с ненормализованными временными рядами.
- 2 “Гауссово ядро”. Этот алгоритм схож с алгоритмом “Евклидово ядро” за исключением того, что он использует Гауссово ядро (2.2) вместо евклидова (2.1).
- 3 “Гауссово ядро, нормализованные временные ряды”. Этот алгоритм аналогичен алгоритму “Гауссово ядро”, но работает с Z-нормализованными временными рядами. Z-нормализованный временной ряд получается из исходного путем применения следующего преобразования к каждому элементу ряда:

$$\mathbf{s}'_i = \frac{\mathbf{s}_i - \bar{\mathbf{s}}}{\sqrt{\frac{1}{n} \sum_j (\mathbf{s}_j - \bar{\mathbf{s}})^2}}, \quad \bar{\mathbf{s}} = \frac{1}{n} \sum_j \mathbf{s}_j, \quad i \in \{1, \dots, n\}.$$

- 4 “Гауссово ядро, нормализованные временные ряды, удаление концов пути”. В дополнение к алгоритму “Гауссово ядро, нормализованные временные ряды” концы построенного выравнивающего пути обрезаются. Концы оптимального пути из-за граничных условий добавляют большие значения и влияют на распределение элементов множества отклонений.

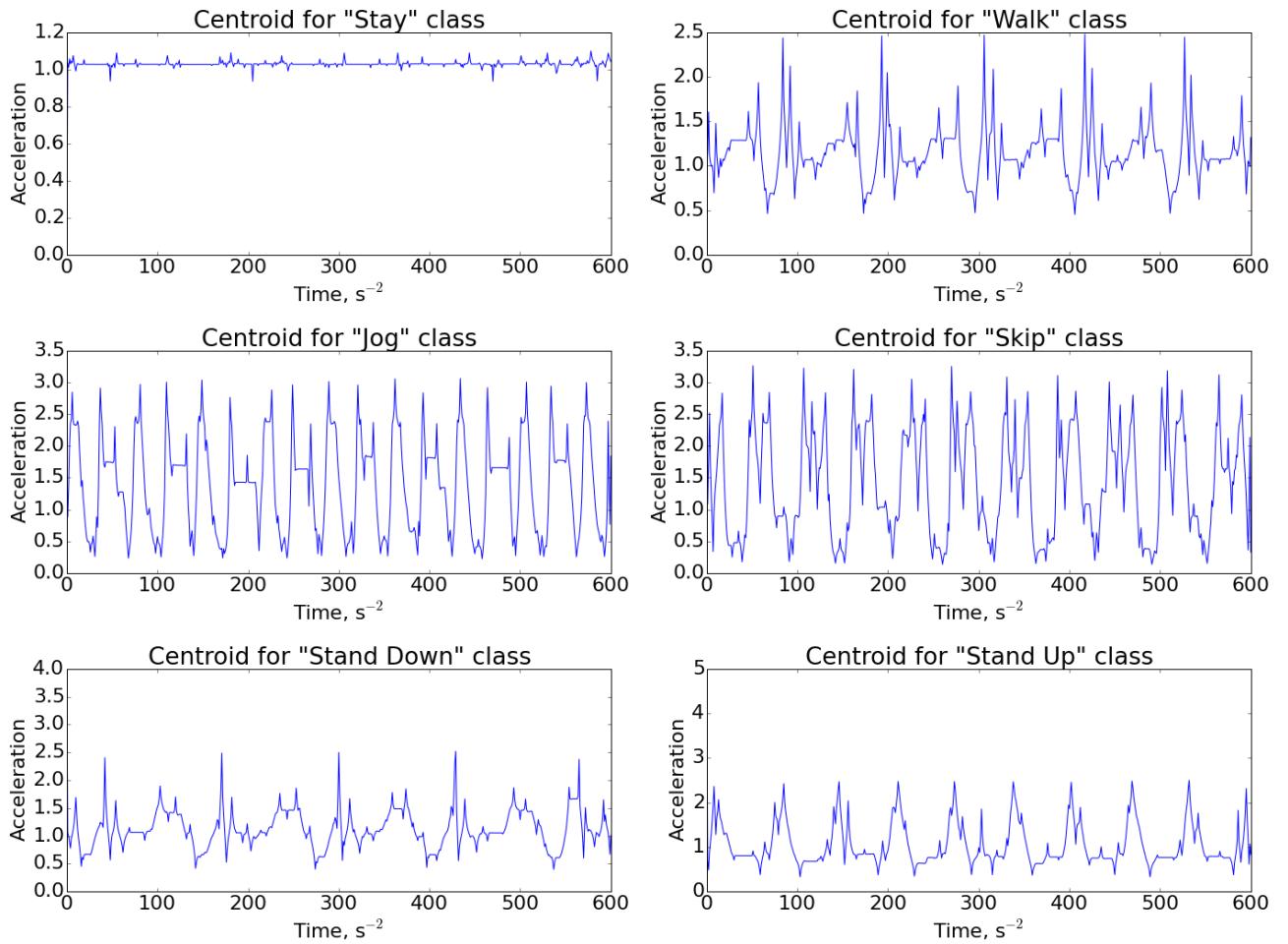


Рис. 6.10. Центроиды для шести типов физической активности человека

Полоса Сакое-Чиба задает глобальные ограничения на положение оптимального пути. Это уменьшает вычислительную сложность и улучшает качество классификации. Для любого элемента (i, j) выравнивающего пути $\hat{\pi}$ и константы k выполняется $i - k \leq j \leq i + k$. Пусть константа k зафиксирована. Тогда вычислительная сложность построения оптимального пути уменьшается от $O(n^2)$ до $O(n)$.

В эксперименте сравнивается качество классификации для двух различных способов использования множества отклонений δ

- 1 “Усреднение”. Среднее значение (5.17) множества δ принимается за функцию расстояния ρ при фиксированных \mathbf{a} и \mathbf{d} .
- 2 “Эвристика”. δDTW (5.18) принимается за функцию расстояния ρ при фиксированных \mathbf{a} и \mathbf{d} . Признаковое пространство строится для вектора α :

$$\alpha = [0, 0.1, \dots, 1.0], \quad \text{с шагом } 0.1.$$

В таблице 6.5 показаны результаты классификации для трех разных моделей классификации f : случайный лес (Random forest), SVM, KNN.

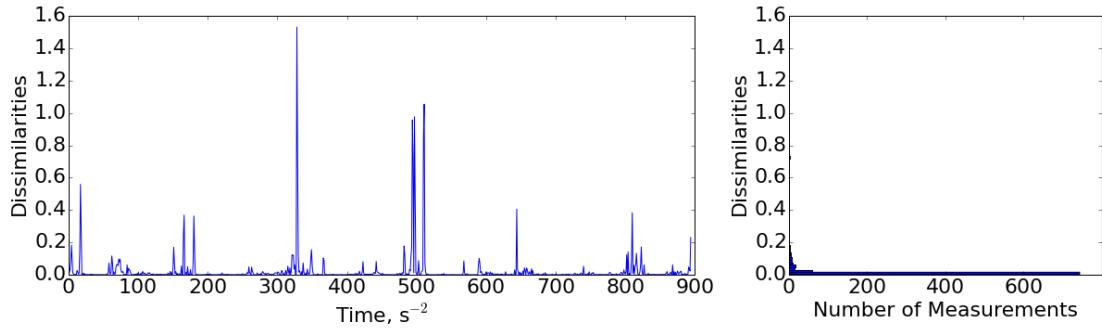


Рис. 6.11. Множество отклонений для евклидова ядра. а) упорядоченное во времени множество. б) Гистограмма распределения.

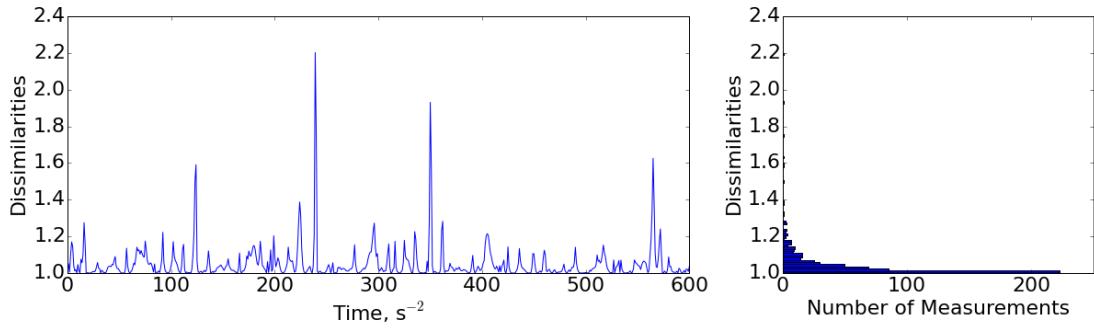


Рис. 6.12. Множество отклонений для гауссова ядра. а) упорядоченное во времени множество. б) Гистограмма распределения.

Комбинация модели Random forest, алгоритма а “Гауссово ядро, нормализованные временные ряды, удаление концов пути” и предложенной функции расстояния δ DTW дает наилучшую точность классификации. Линейный классификатор не показывает хорошие результаты. Метод ближайших соседей в комбинации с функцией расстояния “усреднение” достигает более хороших результатов. Это объясняется интерпретируемостью такого подхода. Модель Random forest дает высокое качество классификации: она превосходит другие методы в среднем на 7%.

Таблица 6.5. Качество многоклассовой классификации временных рядов методами Random forest, SVM, KNN

| | Random forest | | SVM | | KNN | |
|---|----------------------|-------------|------------|-------------|-------------|-----------|
| | Усреднение | Эвристика | Усреднение | Эвристика | Усреднение | Эвристика |
| Евклидово ядро | 0.71 | 0.82 | 0.55 | 0.60 | 0.68 | 0.62 |
| Гауссово ядро | 0.77 | 0.84 | 0.39 | 0.60 | 0.76 | 0.65 |
| Гауссово ядро, Нормализованные временные ряды | 0.78 | 0.85 | 0.56 | 0.60 | 0.78 | 0.60 |
| Гауссово ядро, Нормализованные временные ряды, Удаление концов пути | 0.82 | 0.86 | 0.56 | 0.63 | 0.81 | 0.70 |

6.5. Вычислительный эксперимент по метрической многоклассовой классификации и классификации в пространстве временных рядов (магистерская диссертация Моргачева)

В этом эксперименте ставились задача метрической многоклассовой классификации методом k ближайших соседей и задача классификации в пространстве представлений методом опорных векторов. Использовались алгоритмы выравнивания без аппроксимации (DTW), аппроксимации с помощью полиномов степеней 1 и 4 (DTW_{1p} и DTW_{4p}), а также алгоритм монотонной нейронной сети (NN). В роли шаблонов выступали центроиды классов, а в роли критерия качества Q – точность, вычисленная по формуле:

$$Q = \frac{1}{N} \sum_{i=1}^N [f_p(x_i) = y_i]$$

В эксперименте участвовали три набора данных с различными длинами временных рядов:

1. Распознавание активностей человека по показаниям акселерометра телефона UCI HAR [57]. В этом датасете содержатся временные ряды, соответствующие шести активностям человека: ходьба, подъём по лестнице, спуск по лестнице, сидение, стояние, лежание (21 измерение).
2. Распознавание жестов по показаниям акселерометра контроллера Wii [58]. Временные ряды данного набора данных относились к 10 классам: взятие предмета, тряска, движение вправо, движение влево, движение вверх, движение вниз, круговое движение по часовой стрелке, круговое движение против часовой стрелки, движение за пределы экрана, движение обратно на экран (100 измерений).
3. Данные об использовании электроэнергии в домах [59]. Временные ряды соответствуют потреблению энергии при различных комбинациях электрических приборов, использующихся в доме (80 измерений).

Все выборки были разбиты на обучающую и тестовую части. Каждый ряд был нормализован:

$$x_n = \frac{x - E[x]}{Var[x]}.$$

Можно заметить, что на задаче классификации временных рядов с помощью метода k ближайших соседей (таблица 6.6) на всех трех датасетах наибольшую точность, как и ожидалось, имел алгоритм DTW без аппроксимации.

Аппроксимации с помощью полиномов степеней 1 и 4 показывают близкие результаты на обеих задачах, что может быть объяснено сложностью обучения более сложной функции с помощью генетического алгоритма (нужно ли здесь пояснить о генетическом алгоритме подробнее?).

Таблица 6.6. Результаты классификации временных рядов для задач классификаций активности человека, жестов и электроприборов

| Функция расстояния | Скорость, ит./с | Q_1 |
|------------------------------------|-----------------|--------------------|
| Распознавание активностей человека | | |
| DTW | 19.0 | 0.71 ± 0.02 |
| DTW_{p1} | 12.8 | 0.53 ± 0.06 |
| DTW_{p4} | 13.5 | 0.56 ± 0.04 |
| NN | 90.2 | 0.51 ± 0.06 |
| Распознавание жестов | | |
| DTW | 2.2 | 0.46 ± 0.04 |
| DTW_{p1} | 4.4 | 0.43 ± 0.06 |
| DTW_{p4} | 4.3 | 0.41 ± 0.04 |
| NN | — | — |
| Распознавание электроприборов | | |
| DTW | 2.4 | 0.71 ± 0.02 |
| DTW_{p1} | 2.2 | 0.40 ± 0.06 |
| DTW_{p4} | 2.2 | 0.29 ± 0.04 |
| NN | — | — |

Эксперимент с монотонной нейронной сетью (NN) проведен только для первого датасета, так как эта архитектура позволила работать только с временными рядами малой длины. В сравнении с DTW время работы было меньше, но ввиду отсутствия результатов на остальных наборах данных, сравнение NN с остальными подходами было бы некорректно.

Таблица 6.7. Результаты классификации в пространстве представлений для задач классификаций активности человека, жестов и электроприборов

| Функция расстояния | Скорость, ит./с | Q_2 |
|------------------------------------|-----------------|--------------------|
| Распознавание активностей человека | | |
| DTW | 18.2 | 0.61 ± 0.05 |
| DTW_{p1} | 12.2 | 0.48 ± 0.04 |
| DTW_{p4} | 10.3 | 0.51 ± 0.02 |
| NN | 90.0 | 0.41 ± 0.06 |
| Распознавание жестов | | |
| DTW | 1.8 | 0.38 ± 0.04 |
| DTW_{p1} | 2.3 | 0.40 ± 0.03 |
| DTW_{p4} | 2.3 | 0.37 ± 0.03 |
| NN | — | — |
| Распознавание электроприборов | | |
| DTW | 2.2 | 0.44 ± 0.11 |
| DTW_{p1} | 2.1 | 0.30 ± 0.02 |
| DTW_{p4} | 2.2 | 0.26 ± 0.09 |
| NN | — | — |

Для коротких временных рядов алгоритм DTW без аппроксимации показывает гораздо меньшее время вычисления функции расстояния, чем подходы на основе аппроксимации полиномами (18 ит./с против 10 ит./с) на датасете №1 [57] в задаче построения шаблона (таблица 6.7). На длинных же временных рядах быстрее работают аппроксимирующие алгоритмы.

На датасете №2 [58] алгоритмы динамического выравнивания и аппроксимированного выравнивания показывают схожие результаты. Это может быть объяснено структурой данных: на рисунке 6.13 видно, что матрица попарных расстояний между рядами одного класса для этого датасета содержит в себе пути выравнивания простой структуры, который легко находится предлагаемой моделью аппроксимации.

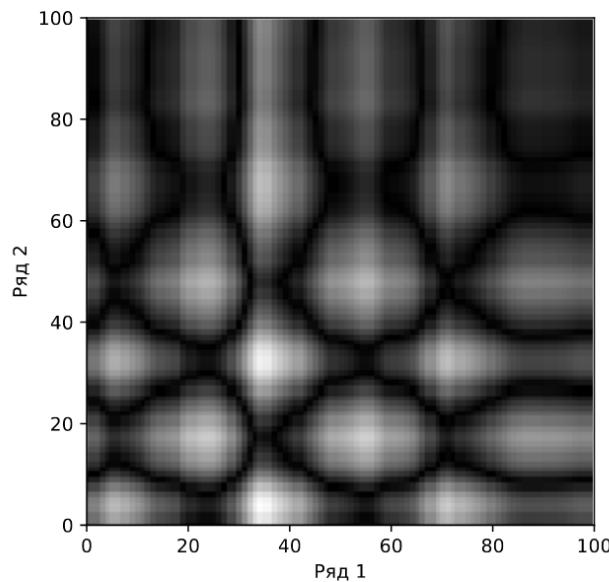


Рис. 6.13. Матрица попарных расстояний между точками для распознавания жестов.

Можно заметить, что на наборе данных №3 [59] динамическое выравнивание показало значительно лучшие результаты, чем аппроксимирующие аналоги. Это может быть следствием особенности структуры временных рядов: в этом датасете представители различных классов имеют схожую структуру и для их различия мало подходят аппроксимирующие алгоритмы. Примеры этих временных рядов можно увидеть на рисунке 6.14.

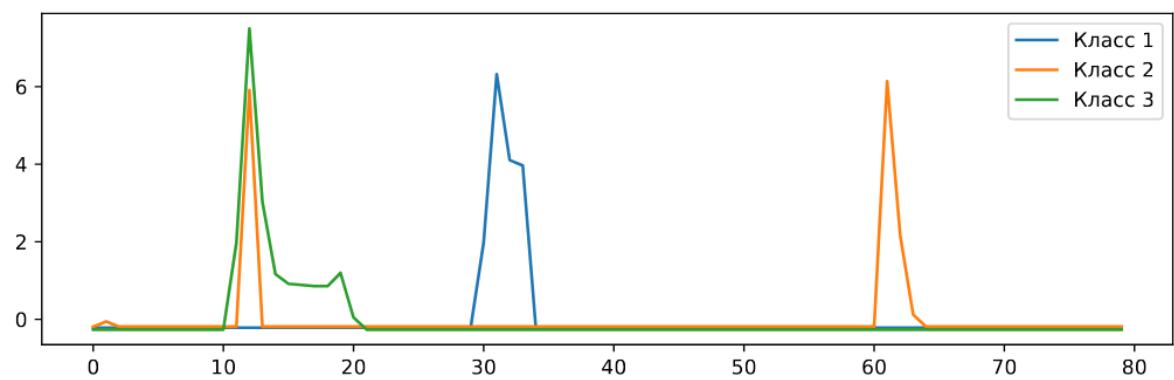


Рис. 6.14. Примеры объектов из датасета используемых электроприборов.

6.6. Вычислительный эксперимент, подтверждающий корректность применения метода DTW к пространственно-временным объектам

Постановка задачи

В данном эксперименте исследуется корректность применения метода DTW и его модификаций к пространственно-временным рядам. Проверяется, что функция, создаваемая алгоритмом динамического выравнивания, является ядром. Проверка этого факта осуществляется при помощи теоремы Мерсера, основная часть которой заключается в проверке матрицы попарных расстояний на неотрицательную определенность. Также производится анализ зависимости качества классификации методом опорных векторов и методом k-ближайших соседей от различных функций.

Проверяется выполнение условий теоремы Мерсера на разных данных для разных модификаций DTW. То есть, следующие два условия для функции $K(x, x')$, порожденной DTW:

- $K(x, x') = K(x', x)$
- $\int_X \int_X K(x, x') g(x) g(x') dx dx' \geq 0 \quad \forall g : X \rightarrow \mathbb{R}$

Последнее условие эквивалентно тому, что для любых наборов $\{x_1, \dots, x_n\}$ матрица $K = \|K(x_i, x_j)\|_{i,j}$ неотрицательно определена: $v^T K v \geq 0 \quad \forall v \in \mathbb{R}^n$. В нашей задаче мы будем исследовать, насколько качественно функции (являющиеся ядрами или нет), полученные в результате DTW, подставленные в алгоритм SVM (Support Vector Machine), классифицируют объекты. Для начала рассмотрим задачу классификации объектов $X \in \mathbb{R}^n$ на два непересекающихся класса $Y = \{-1, +1\}$. Обучающая выборка $X^l = (x^j, y^j)_{j=1}^l$. Линейный классификатор будет иметь вид:

$$a(x) = \text{sign}\left(\sum_{i=1}^n \omega_i x_i - \omega_0\right) = \text{sign}(\langle \omega, x \rangle - \omega_0)$$

Тут использованы обозначения: $x = (x_1, \dots, x_n)$ – признаковое описание объекта x , $\omega = (\omega_1, \dots, \omega_n) \in \mathbb{R}^n$ и $\omega_0 \in \mathbb{R}$ являются, так называемыми весами, являющимися параметрами алгоритма. Заметим, что $\langle \omega, x \rangle = \omega_0$ задает в пространстве гиперплоскость, которая разделяет классы.

Заметим, что линейный классификатор $a(x)$ не изменится если ω и ω_0 умножить на одну и ту же положительную константу. Поэтому произведем нормировку наиболее удобным для нас способом: чтобы для всех ближайших к разделяющих гиперплоскости объектов $x^j \in X^l \hookrightarrow \langle \omega, x^j \rangle - \omega_0 = y^j$. Таким образом, получим для всех $x^j \in X^l$:

$$\langle \omega, x^j \rangle - \omega_0 \begin{cases} \leq -1, & \text{if } y^j = -1 \\ \geq 1, & \text{if } y^j = +1 \end{cases}$$

$-1 < \langle \omega, x \rangle - \omega_0 < 1$ задает полосу, которая разделяет классы. Внутри этой полосы нет ни одной точки обучающей выборки X^l , на ее границе лежат точки, ближайшие к разделяющей гиперплоскости. Сама гиперплоскость проходит ровно посередине полосы. Для более качественной классификации необходимо добиться максимальной ширины h этой полосы.

Пусть точки x_- , x_+ – соответственно точки классов $-1, +1$, лежащие на границе полосы, тогда:

$$h = \frac{1}{\|\omega\|} \langle x_+ - x_-, \omega \rangle = \frac{\langle x_+, \omega \rangle - \langle x_-, \omega \rangle}{\|\omega\|} = \frac{(\omega_0 + 1) - (\omega_0 - 1)}{\|\omega\|} = \frac{2}{\|\omega\|}$$

Таким образом, нам необходимо решить следующую задачу оптимизации:

$$\begin{cases} \frac{1}{2} \langle \omega, \omega \rangle \rightarrow \min \\ (\langle \omega, x^j \rangle - \omega_0) y^j \geq 1 \quad i = 1, \dots, l \end{cases}$$

Решаем данную задачу при помощи теоремы Каруши-Куна Таккера:

$$\begin{cases} L(\omega, \omega_0, \lambda) = \frac{1}{2} \langle \omega, \omega \rangle - \sum_{j=1}^l \lambda_j ((\langle \omega, x^j \rangle - \omega_0) y^j - 1) \rightarrow \min_{\omega, \omega_0} \max_{\lambda} \\ \lambda \geq 0 \\ \lambda_j (\langle \omega, x^j \rangle - \omega_0) = 0 \quad j = 1, \dots, l \end{cases}$$

$\lambda = (\lambda_1, \dots, \lambda_l)$ – вектор двойственных переменных.

Необходимым условием седловой точки Лагранжиана L является равенство нулю его градиента, отсюда получаем:

$$\omega = \sum_{j=1}^l \lambda_j y^j x^j, \quad \sum_{j=1}^l \lambda_j y^j = 0 \quad (6.2)$$

Из (6.2) следует, что вектор весов ω является линейной комбинацией таких векторов обучающей выборки X^l , для которых соответствующее $\lambda_j \neq 0$. Согласно условию дополняющей нежесткости, так как $\lambda_j \neq 0$, исходные ограничения типа неравенств должны превратиться в равенства, следовательно, эти объекты (векторы) находятся на границе полосы. Векторы, для которых $\lambda_j = 0$, не лежат на границе и не участвуют в сумме, следовательно, если бы их не было в выборке, алгоритм не изменился.

Объект $(x^j, y^j) \in X^l$, для которого $\lambda_j > 0$ и $\langle \omega, x^j \rangle - \omega_0 = y^j$, назовем опорным вектором (англ. support vector).

Подставим (6.2) обратно в выражение для Лагранжиана, получим:

$$\begin{cases} -L = -\sum_{j=1}^l \lambda_j + \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \lambda_i \lambda_j y^i y^j \langle x^i, x^j \rangle \rightarrow \min_{\lambda} \\ \lambda \geq 0 \\ \sum_{j=1}^l \lambda_j y^j = 0 \end{cases} \quad (6.3)$$

Полученная задача имеет единственное решение, поскольку целевая функция является квадратичным функционалом, имеющим неотрицательно определенную квадратичную форму, следовательно, является выпуклым, ограничения также выпуклы.

После решения задачи мы можем определить вектор ω по формуле $\omega = \sum_{j=1}^l \lambda_j y^j x^j$ и $\omega_0 = \text{med}\{\langle \omega, x^j \rangle - y^j : \lambda_j > 0, j = 1, \dots, l\}$.

Тогда получим: $a(x) = \text{sign}(\sum_{j=1}^l \lambda_j y^j \langle x^i, x^j \rangle - \omega_0)$.

Вопрос о решении двойственной задачи (6.3) все еще остается открытым.

Все эти рассуждения имеют место в случае, когда выборка линейно разделима, если же она не является таковой, то необходимо перейти в пространство большей размерности, где уже она будет линейно разделима. Этот переход будет осуществляться засчет функции ядра. Скалярные произведения $\langle x, x' \rangle$, таким образом, везде заменяются на значение функции ядра в соответствующих двух точках $K(x, x')$.

Эксперимент

Производится тестирование различных модификаций алгоритма DTW на различных данных [60], затем осуществляется проверка того, является ли полученная в результате работы алгоритма функция ядром (при помощи теоремы Мерсера).

Строятся выравнивающие пути попарно между пятью временными рядами:

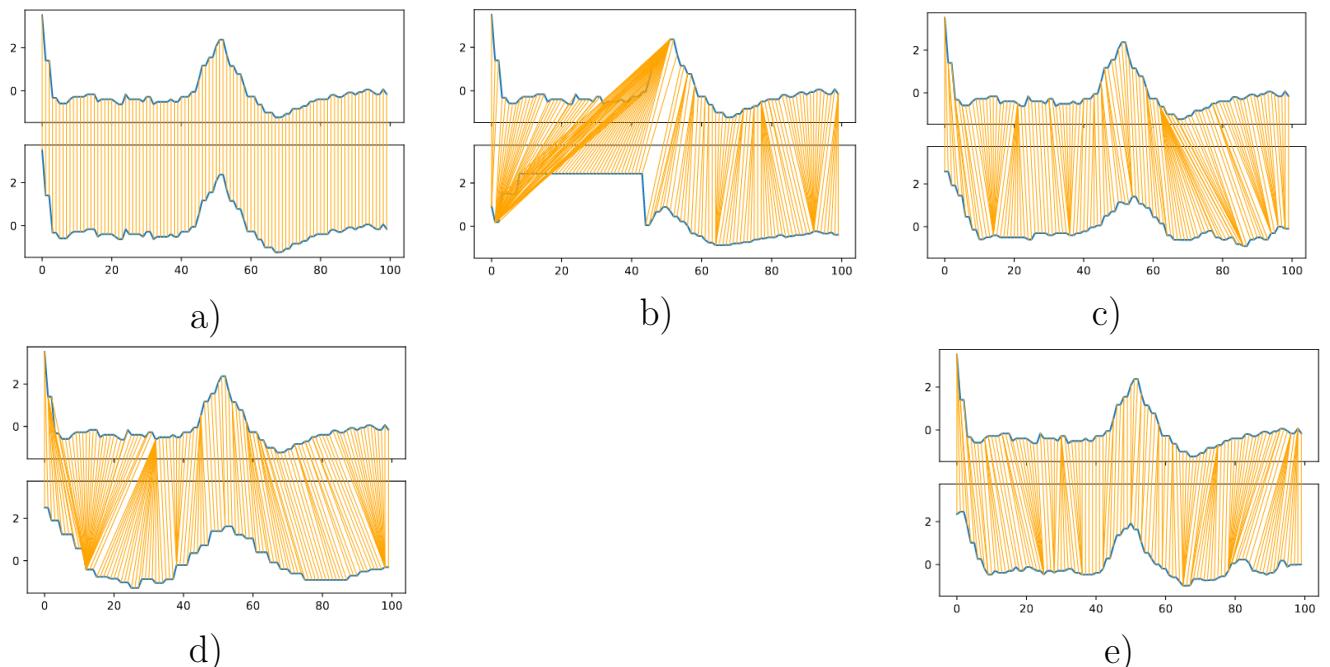
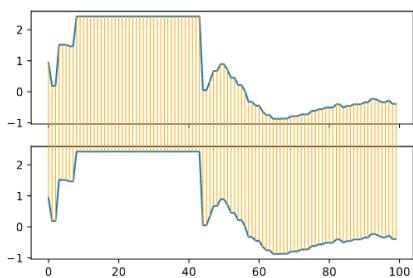
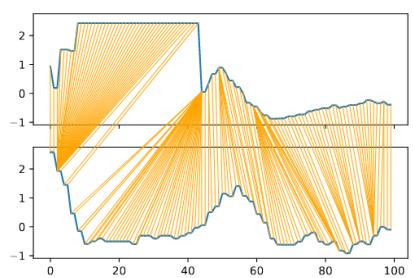


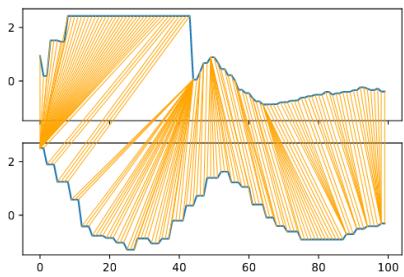
Рис. 6.15. Выравнивание попарно пяти временных рядов: а) 1-1, б) 1-2, в) 1-3, г) 1-4, д) 1-5.



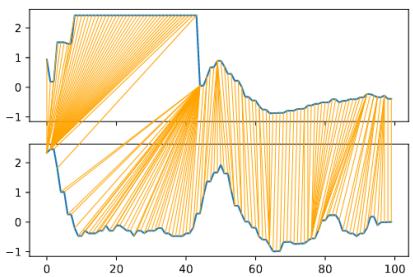
f)



g)

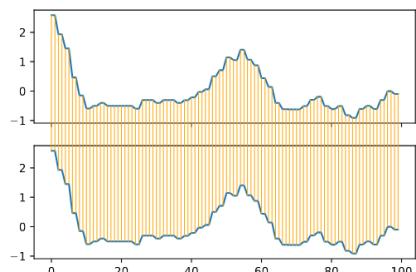


h)

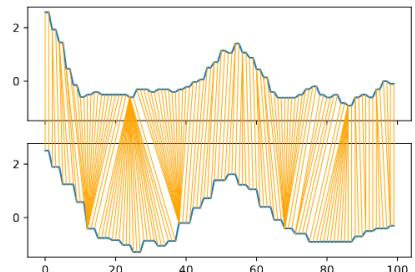


i)

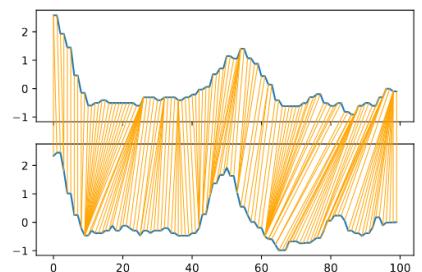
Рис. 6.16. Выравнивание попарно пяти временных рядов (продолжение): f) 2-2, g) 2-3, h) 2-4, i) 2-5.



j)

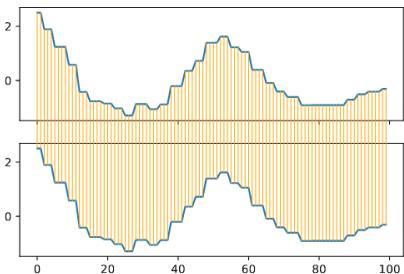


k)

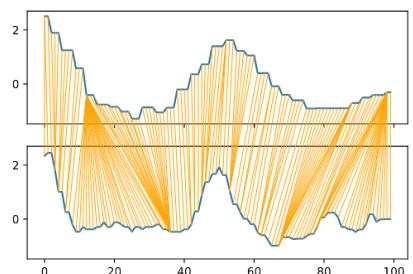


l)

Рис. 6.17. Выравнивание попарно пяти временных рядов (продолжение): j) 3-3, k) 3-4, l) 3-5.



m)



n)

Рис. 6.18. Выравнивание попарно пяти временных рядов (продолжение): m) 4-4, n) 4-5.

С помощью DTW посчитана матрица D попарных расстояний между рядами:

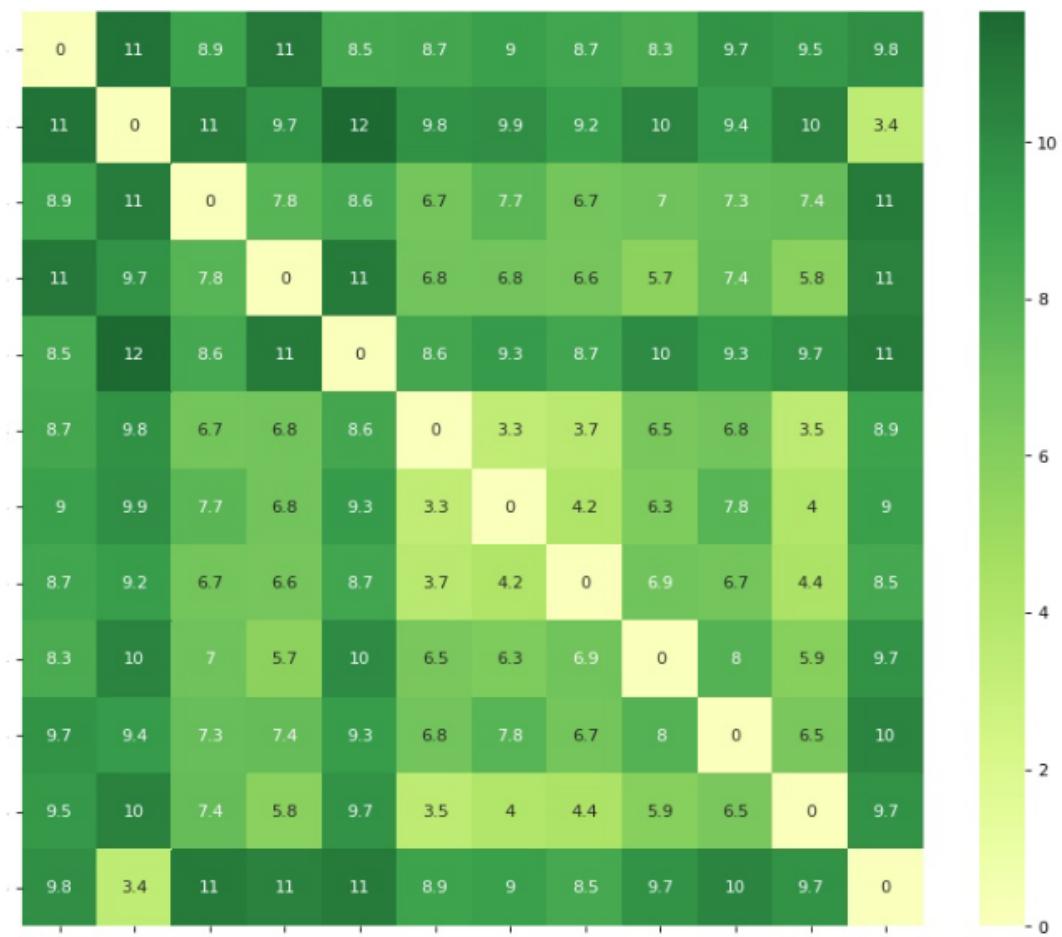
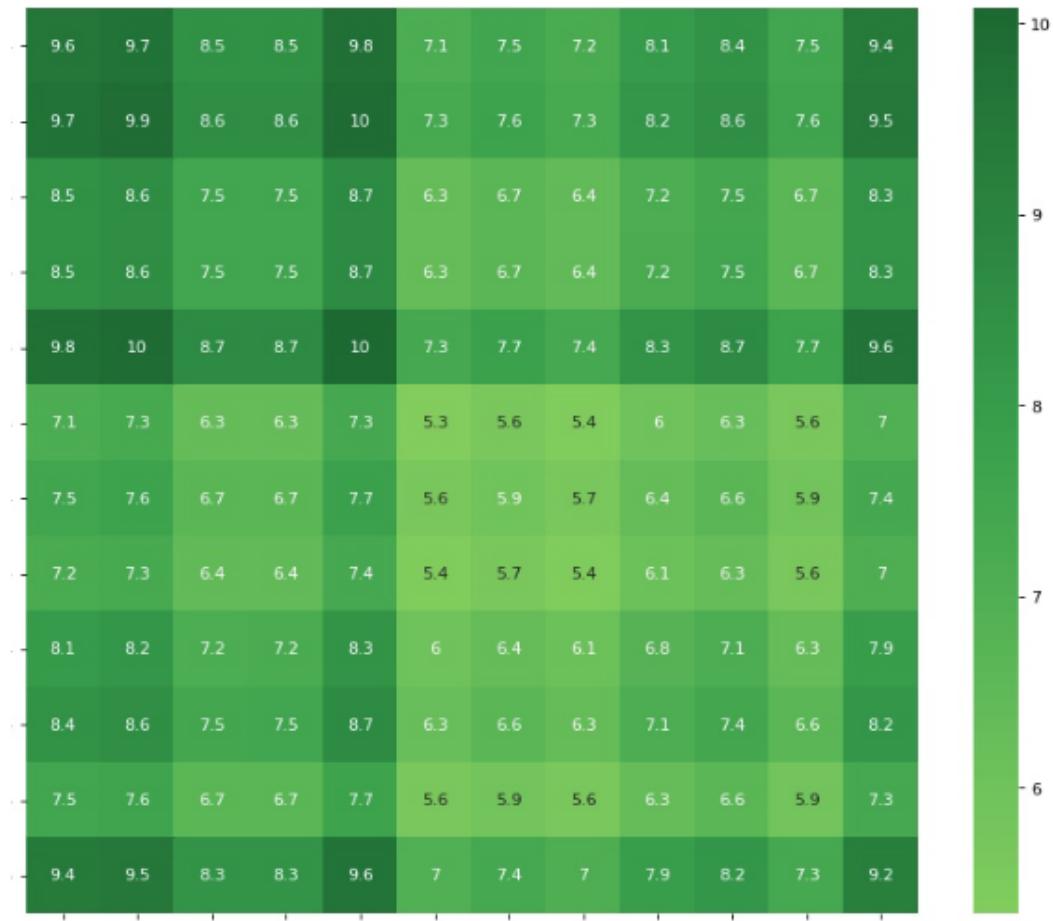


Рис. 6.19. Матрица D .

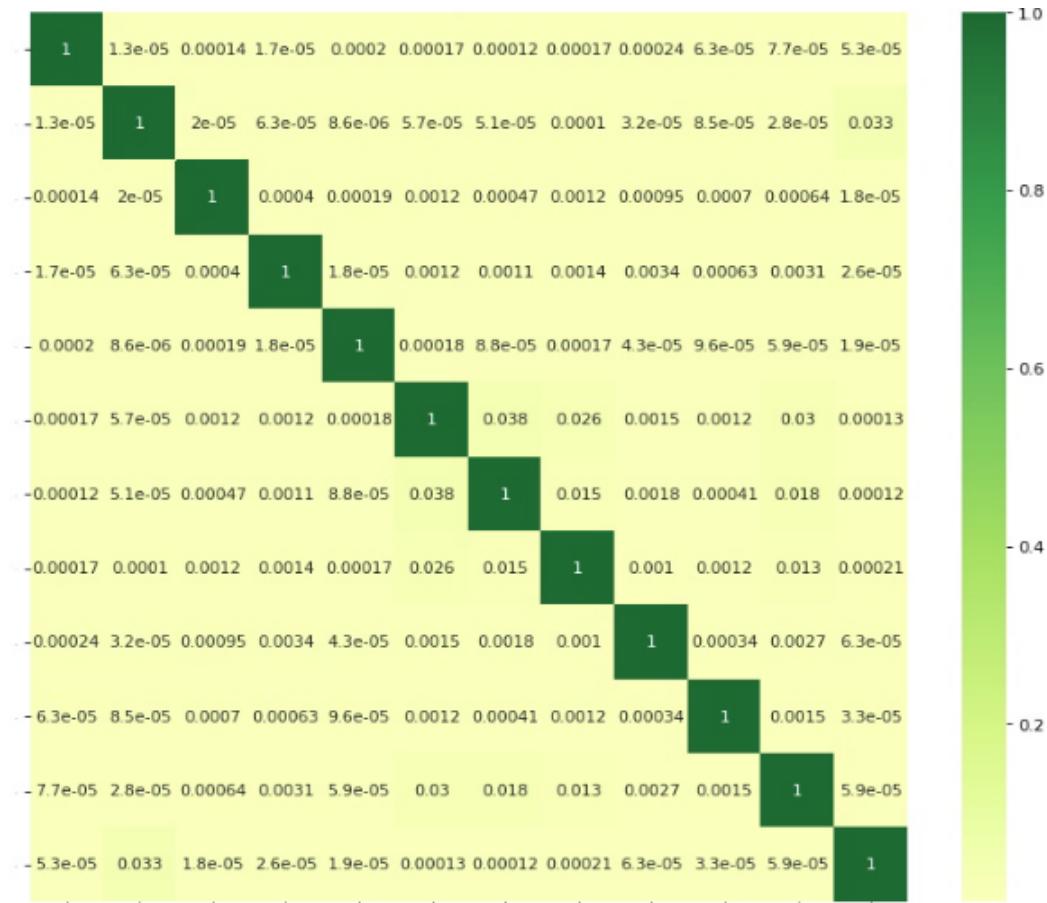
Матрица D не является положительно полуопределенной (\Rightarrow не является ядром), поэтому мы будем искать ближайшую к ней положительно полуопределенную матрицу A следующим образом:

Так как матрица D - симметричная, можно разложить ее в виде $D = QMQ^T$, где матрица M - диагональная, причем на диагонали стоят собственные значения матрицы D . Обозначим $M_+ = \max(M, 0)$, тогда $A = QM_+Q^T$. Матрица A - положительно полуопределенная, так как все ее собственные значения – элементы диагональной матрицы M_+ – неотрицательны.

Рис. 6.20. Матрица A .

Эксперимент подтверждает, что матрица A является положительно полуопределенной.

Также можно создавать RBF ядро $K(x, x') = \exp(-\gamma \times \rho_{DTW}(x, x'))$, где $\rho_{DTW}(x, x')$ – расстояние между рядами x и x' . Полученная матрица Z также получится неотрицательно определенной.

Рис. 6.21. Матрица Z .

Затем, мы считаем таким же образом матрицы D , A , Z для временных рядов большего размера и используем их, а также просто норму L_2 в качестве "ядра" в алгоритме SVM для их классификации.

И, замерив метрику качества, получаем следующие результаты:

- для L_2 : 0,182
- для D (просто DTW): 0,327
- для A (ближайшая к D неотрицательно определенная): 0,523
- для Z (RBF ядро): 0,864

Как и ожидалось, метрика L_2 дала худший результат, следующий результат дала матрица D , улучшив предыдущий на 20%, затем матрица A – аппроксимация D , улучшив результат еще на 20%, и лучший результат дало RBF ядро Z .

6.7. Вычислительный эксперимент по подбору оптимальной функции расстояния для кластеризаций многомерных временных рядов

Постановка задачи

См. постановку задачи в главе 3.2 "Выравнивание многомерных рядов" и в главе про DTW

Также см. задачу поиска подпоследовательности в главе 3.2

Постановка задачи кластеризации

Suppose we have $\mathbf{X}, |\mathbf{X}| = N$ from a dataset of multivariate time-series of length n . Besides, there exists a label $y \in \mathbf{Y}$ for all $X \in \mathbf{X}$, where \mathbf{Y} is the set of class labels. Lets define a distance matrix between *pairs of time-series*:

$$M(DTW_\rho(\mathbf{X})) = \|M_{ij}\|, M_{ij} = DTW_\rho(X, Z), X, Z \in \mathbf{X}.$$

Denote by f_d the clustering model:

$$f_d : M \rightarrow \mathbf{S}^N,$$

where \mathbf{S} is the set of cluster labels and d set up the clustering linkage criteria for clusters $A, B \in \mathbf{S}$:

- 1) complete : $d(A, B) = \max_{a \in A, b \in B} (dist(a, b));$
- 2) weighted : $d(A, B) = \frac{(dist(S, B) + dist(T, B))}{2}$, here cluster $A = S \cup T$;
- 3) average : $d(A, B) = \sum_{a \in A, b \in B} \frac{d(a, b)}{(|A| * |B|)}.$

We consider the next quality metrics:

$$Q_{clustering1}(f, \mathbf{X}, \rho) = \frac{1}{|\mathbf{S}|} \sum_{s \in \mathbf{S}} \max_y \frac{N_s^y}{N_s},$$

$$Q_{clustering2}(f, \mathbf{X}, \rho) = \frac{1}{|\mathbf{S}|} \sum_{s \in \mathbf{S}} \max_y \frac{(N_s^y)^2}{N_s N^y},$$

where N_s is the number of elements in the cluster labeled by s , N^y is the number of elements in the class with label y , N_s^y is the number of elements such that they contain class label y and cluster label s .

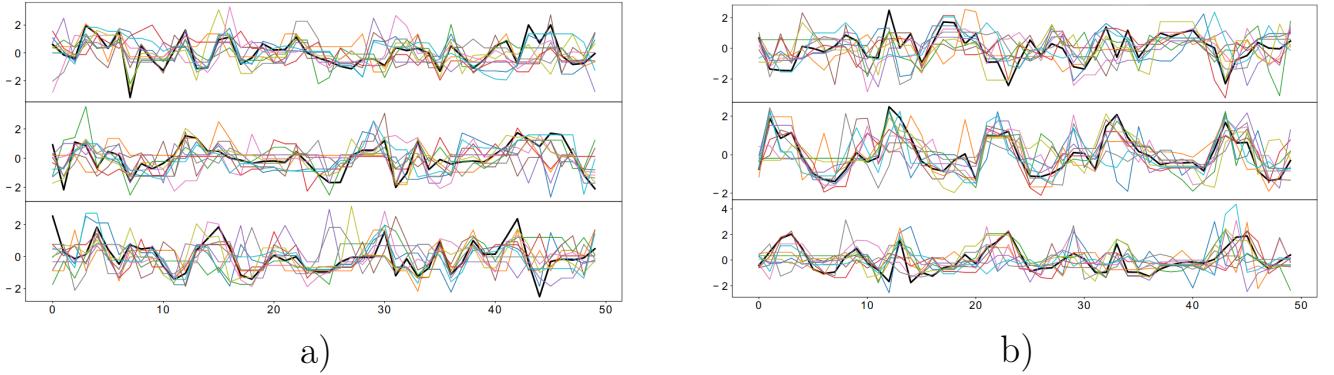


Рис. 6.22. Clustering of 3-dimensional time-series: a) with L_1 based DTW, b) with L_2 based DTW.

Эксперимент

The paper contains the comparing of the approach quality with quality of another ways to solve model tasks:

- 1) a pairwise approach to define distance function between multivariate time-series:

$$\text{ED}(X, Y) = \sum_{i=1}^T \|x_t - y_t\|_2,$$

- 2) representation of a time-series via coefficients \mathbf{a} of the autoregression model:

$$x_t = c + \sum_{i=1}^p a_i x_{t-i} + \varepsilon_t.$$

Such coefficients are considered as features representation of original time-series. The L_2 metric is used on space of features and apply the same metric methods to solve the considered tasks.

Subsequence Search The experiment uses two dataset [61; 62] to evaluate metrics. The first is a data from graphics tables represented by three-dimensional time-series with length 182. Two channels belong to the coordinates of a pen and one belongs to the pressing force. The second sample consists of marked accelerometer data whilst conducting 4 different activities: gesturing, walking slowly, walking normal and walking fast. Time-series in the dataset are three dimensional with length 206. The main difference in these datasets is the repetition period of patters. So, while the first dataset has only one repetition per length, the second dataset has 5 - 6 repetitions per length.

For both sample, a random subset of representatives is chosen to create the patterns via DBA [63] and simple averaging methods. The result patterns are shown in ??.

The others time-series are united in random order into a one long time-series. Then for all patterns we found k nearest subsequences of the long time-series in the meaning of ED and DTW_ρ with different choices of distance function ρ . To estimate error, the described experiment has been conducted for 20 samples, which are generated by *bootstrap* [64].

| ρ | average | characters | | | epi | | |
|----------|---------|----------------|----------------|-----------------|----------------|------------|-----------------|
| | | Q | t | $t_{no\ optim}$ | Q | t | $t_{no\ optim}$ |
| L_1 | DBA | 84 ± 14 | $2,56 \pm 0,4$ | 11 ± 1 | 54 ± 10 | 26 ± 1 | 25 ± 1 |
| | mean | 87 ± 12 | $2,77 \pm 0,4$ | 14 ± 1 | 49 ± 08 | 26 ± 1 | 28 ± 2 |
| L_2 | DBA | 81 ± 16 | $2,24 \pm 0,3$ | 13 ± 1 | 55 ± 07 | 23 ± 1 | 26 ± 2 |
| | mean | 84 ± 13 | $2,17 \pm 0,2$ | 10 ± 1 | 48 ± 10 | 23 ± 1 | 26 ± 2 |
| cos_dist | DBA | 79 ± 17 | $3,42 \pm 1,0$ | 16 ± 1 | 35 ± 10 | 29 ± 2 | 38 ± 2 |
| | mean | 80 ± 15 | $3,43 \pm 0,8$ | 14 ± 1 | 27 ± 06 | 26 ± 2 | 38 ± 1 |
| ED | DBA | 8 | 17,511 | 17,511 | 17,2 | 1,620 | 1,620 |
| | mean | 9 | 17,645 | 17,645 | 17,2 | 1,540 | 1,540 |

Таблица 6.8. Subsequence Search.

The results of the experiments are shown in 6.8. So, the experiment shows that L_1 based cost function gives slightly better quality rate than the others in all cases except search DBA averaged patterns in second dataset, when L_1 show quality rate in 54% and L_2 give 55%. The L_1 based cost function gives similar results and the cost function with cosine distance between values gives the worst quality everywhere.

So, the optimization allow us to speed up five times subsequence search for the character dataset, but the search time for the second dataset remains the same.

Clustering The experiment uses origin dataset with three-dimensional time-series from accelerometer. The test data contains 20 samples of 600 time-series with length 50 such that all samples are generated from the original dataset by bootstrap. All datasets have one of four class label. Besides, samples are uniformly distributed across classes.

The experiment calculates the distance matrices between pairs of time-series for ED and DTW with different choices of distance function ρ . Then the matrices are used to cluster time-series via hierarchical clustering methods with the described linkage criteria.

Besides, an autoregression model is trained for each time-series; then we cluster the coefficients vector of the pre-trained model using hierarchical clustering.

Because of different accelerometer position during data collection, we consider the next number of clusters: 4, 12, 24, 36, 48.

| ρ | N_{clus} | Q_1 | | | Q_2 | | |
|----------|------------|------------|------------|------------------------------|-------------|------------|------------------------------|
| | | complete | average | weighted | complete | average | weighted |
| L_1 | 4 | 70 \pm 7 | 75 \pm 5 | 80 \pm 1 | 59 \pm 10 | 68 \pm 7 | 76 \pm 1 |
| | 12 | 89 \pm 3 | 91 \pm 3 | 89 \pm 1 | 85 \pm 4 | 89 \pm 4 | 86 \pm 2 |
| | 24 | 92 \pm 2 | 94 \pm 1 | 95 \pm 1 | 88 \pm 3 | 92 \pm 2 | 93 \pm 1 |
| | 36 | 92 \pm 2 | 95 \pm 1 | 96 \pm 1 | 88 \pm 2 | 92 \pm 2 | 94 \pm 1 |
| | 48 | 92 \pm 1 | 95 \pm 1 | 97 \pm 1 | 89 \pm 2 | 93 \pm 2 | 95 \pm 1 |
| L_2 | 4 | 73 \pm 7 | 76 \pm 2 | 80 \pm 1 | 64 \pm 11 | 69 \pm 4 | 76 \pm 1 |
| | 12 | 84 \pm 4 | 92 \pm 2 | 89 \pm 1 | 84 \pm 5 | 88 \pm 3 | 86 \pm 1 |
| | 24 | 93 \pm 1 | 94 \pm 1 | 95 \pm 1 | 90 \pm 2 | 92 \pm 2 | 93 \pm 1 |
| | 36 | 93 \pm 1 | 95 \pm 1 | 96 \pm 1 | 90 \pm 2 | 93 \pm 1 | 95 \pm 1 |
| | 48 | 93 \pm 1 | 96 \pm 1 | 97 \pm 1 | 90 \pm 1 | 94 \pm 1 | 95 \pm 1 |
| cos_dist | 4 | 57 \pm 8 | 62 \pm 6 | 64 \pm 7 | 43 \pm 10 | 48 \pm 9 | 53 \pm 9 |
| | 12 | 68 \pm 2 | 79 \pm 3 | 82 \pm 4 | 54 \pm 3 | 70 \pm 5 | 74 \pm 6 |
| | 24 | 76 \pm 2 | 86 \pm 2 | 90 \pm 3 | 65 \pm 3 | 79 \pm 3 | 85 \pm 4 |
| | 36 | 80 \pm 1 | 88 \pm 2 | 92 \pm 1 | 70 \pm 2 | 83 \pm 3 | 88 \pm 3 |
| | 48 | 82 \pm 1 | 90 \pm 1 | 92 \pm 1 | 73 \pm 2 | 84 \pm 2 | 88 \pm 1 |
| AR | 4 | 61 \pm 7 | | | 43 \pm 10 | | |
| | 12 | 60 \pm 5 | | | 41 \pm 6 | | |
| | 24 | 68 \pm 3 | | | 52 \pm 5 | | |
| | 36 | 73 \pm 3 | | | 59 \pm 4 | | |
| | 48 | 77 \pm 2 | | | 64 \pm 2 | | |

Таблица 6.9. Clustering.

The clustering results are shown in 6.9. The clustering experiment demonstrates that L_1 and L_2 based cost functions achieve very similar values for both Q_1 and Q_2 quality metrics. The usage of cosine distance is significant decrease quality of clustering: the maximum values of Q_1 and Q_2 are 97% and 95% for L_1 distance function and 92% and 88% for cosine distance. All maximum values are achieved for weighted linkage method. So, the application of autoregression model give the worst results.

Глава 7

Заключение

В разделе 2.2 предложено обобщение метода динамического выравнивания временных рядов для случая объектов, определенных на двух осях времени. Существует теоретическое обобщение предлагаемых методов на случай конечного множества осей времени. Вычислительный эксперимент позволил проанализировать свойства подхода: устойчивость к допустимым преобразованиям и разделяющая способность функции расстояния как на реальных, так и модельных данных. Качество решения задачи метрической классификации выше решения, основанного на Евклидовом расстоянии. Вычислительная сложность метода высокая, что ограничивает его применимость на объектах высокой размерности.

В разделе 2.4 описан новый подход к работе с центроидами временных рядов, построена модель, использующая веса и матрицы центроидов, и показаны ее преимущества перед моделью, использующей классический метод DTW [1]. Продемонстрировано влияние структуры, длины и физического смысла временных рядов на результаты классификации.

В разделе 2.6 был предложен новый подход к анализу непрерывных временных рядов, основанный на функции расстояния DTW между дискретными рядами и имеющей аналогичные свойства, а также имеющей несколько дополнительных свойств устойчивости. Кроме этого, данный подход универсален по отношению к используемому способу аппроксимации. В дальнейшем планируется применить сплайны Безье. Многие методы оптимизации могут быть использованы для ускорения времени вычислений и улучшения процедуры поиска выравнивающего пути. Планируется также провести сравнение изложенного метода с другими на различных датасетах.

В разделе 2.7 предложен новый подход к построению функции расстояния и анализу отклонений между временными рядами для решения задачи классификации. Дополнительная информация о распределении элементов множества отклонений позволяет значительно улучшить точность классификации.

Список литературы

1. Гончаров А. В., Попова М. С., Стрижов В. В. Метрическая классификация временных рядов с выравниванием относительно центроидов классов // Системы и средства информатики. — 2015. — Т. 25, вып. 4. — С. 52—64.
2. Гончаров А. В., Стрижов В. В. Метрическая классификация временных рядов со взвешенным выравниванием относительно центроидов классов // Информатика и её применения. — 2016. — Т. 10, вып. 2. — С. 36—47.
3. Гончаров А. В., Стрижов В. В. Выравнивание декартовых произведений упорядоченных множеств // Информатика и её применения. — 2020. — Т. 14, вып. 1. — С. 31—39.
4. Goncharov A. V., Strijov V. V. Continious Time Series Alignment in Human Actions Recognition // Proceedings of the AINL Fruct Conference. — 2016. — P. 83—86.
5. Гончаров А. В. Выравнивания декартовых произведений упорядоченных множеств mDTW: Программная реализация алгоритма. — 2019. — URL: <https://github.com/Intelligent-Systems-Phystech/PhDThesis/tree/master/Goncharov2019/MatrixDTW/code> (дата обр. 22.07.2019).
6. Goncharov A. V., Strijov V. V. Analysis of Dissimilarity Set Between Time Series // Computational Mathematics and Modeling. — 2018. — Vol. 29, no. 3. — P. 359—366.
7. Hill N. [et al.]. Classifying EEG and ECoG signals without subject training for fast BCI implementation: comparison of nonparalyzed and completely paralyzed subjects // IEEE Transactions on Neural Systems and Rehabilitation Engineering. — 2006. — Vol. 14, no. 2. — P. 183—186.
8. Gardezi J. [et al.]. Mammogram classification using dynamic time warping // Multimedia Tools and Applications. — 2017. — Vol. 77, no. 3. — P. 3941—3962.
9. Varatharajan R. [et al.]. Wearable sensor devices for early detection of Alzheimer disease using dynamic time warping algorithm // Cluster Computing. — 2018. — Vol. 21, no. 1. — P. 681—690.
10. Celebi S. [et al.]. Gesture recognition using skeleton data with weighted dynamic time warping // VISAPP. — Graduate School of Natural, Applied Sciences, Istanbul Sehir University, 2013. — P. 620—625.
11. Gillian N., Knapp R., O'Modhrain S. Recognition Of Multivariate Temporal Musical Gestures Using N-Dimensional Dynamic Time Warping // Proceedings of the International Conference on New Interfaces for Musical Expression. — 2011. — P. 337—342.
12. Hamdy A., Atia A., Mostafa M. Recognizing Driving Behavior and Road Anomaly using Smartphone Sensors // International Journal of Ambient Computing and Intelligence. — 2017. — Vol. 8, no. 3. — P. 22—37.

13. **Bailador G.** [et al.]. Analysis of pattern recognition techniques for in-air signature biometrics // Pattern Recognition. — 2011. — Vol. 44, no. 10/11. — P. 2468–2478.
14. **Sempena S., Maulidevi N., Aryan P.** Human action recognition using Dynamic Time Warping // Proceedings of the 2011 International Conference on Electrical Engineering and Informatics. — Bandung, 2011. — P. 1–5.
15. **Chapaneri S.** Spoken Digits Recognition using Weighted MFCC and Improved Features for Dynamic Time Warping // International Journal of Computer Applications. — 2012. — Vol. 40, no. 3. — P. 6–12.
16. **Namana M.** [et al.]. Emotion Recognition using Dynamic Time Warping Technique for Isolated Words // International Journal of Computer Science Issues. — 2011. — Vol. 8, no. 5. — P. 306–309.
17. **Thiolliere R.** [et al.]. A hybrid dynamic time warping-deep neural network architecture for unsupervised acoustic modeling // Sixteenth Annual Conference of the International Speech Communication Association. — 2015. — P. 3179–3183.
18. **Raghavendra B.** [et al.]. Cardiac arrhythmia detection using dynamic time warping of ECG beats in e-healthcare systems // 2011 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks. — Lucca, 2011. — P. 1–6.
19. **Chang C.-Y.** [et al.]. D^3TW : Discriminative Differentiable Dynamic Time Warping for Weakly Supervised Action Alignment and Segmentation // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. — 2019. — P. 3546–3555.
20. **Yeh C.-C. M.** [et al.]. Online Amnestic DTW to allow Real-Time Golden Batch Monitoring // Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. — 2019. — P. 2604–2612.
21. **Atia A., Shorim N.** Hand Gestures Classification with Multi-Core DTW // The ISC International Journal of Information Security. — 2019. — Vol. 11, no. 3. — P. 91–96.
22. **Basalto N.** [et al.]. Hausdorff clustering of financial time series // Physica A: Statistical Mechanics and its Applications. — 2007. — Vol. 379, no. 2. — P. 635–644.
23. **Gorelick L.** [et al.]. Actions as Space-Time Shapes // Transactions on Pattern Analysis and Machine Intelligence. — 2007. — Vol. 29, no. 12. — P. 2247–2253.
24. **Smyth P.** Clustering sequences with hidden Markov models // Advances in Neural Information Processing Systems. — University of California, Irvine, 1997. — P. 648–654.

25. **Faloutsos C., Ranganathan M., Manolopoulos Y.** Fast Subsequence Matching in Time-Series Databases // SIGMOD International Conference on Management of Data. Vol. 23. — Minneapolis: ACM, 1994. — P. 419–429.
26. **Berndt D., Clifford J.** Using dynamic time warping to find patterns in time series. // Workshop on Knowledge Discovery in Databases, at the 12th Int'l Conference on Artificial Intelligence. Vol. 10. — Seattle, WA, USA, 1994. — P. 359–370.
27. **Keogh E., Ratanamahatana C. A.** Exact indexing of dynamic time warping // Knowledge and Information Systems. — 2005. — Vol. 7, no. 3. — P. 358–386.
28. **Banerjee A., Ghosh J.** Clickstream clustering using weighted longest common subsequences // Proceedings of the web mining workshop at the 1st SIAM conference on data mining. Vol. 143. — 2001. — P. 33–40.
29. **Vlachos M., Gunopulos D., Kollios G.** Discovering similar multidimensional trajectories // Proceedings of the 18th International Conference on Data Engineering. — IEEE Computer Society. San Jose, 2002. — P. 673–684.
30. **Chen L., Ng R. T.** On the marriage of l_p -norms and edit distance // Proceedings of the 30th International Conference on Very Large Data Bases. Vol. 30. — Toronto: Morgan Kaufmann, 2004. — P. 792–803.
31. **Chen L., Özsü M. T., Oria V.** Robust and fast similarity search for moving object trajectories // Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data. — Baltimore: ACM, 2005. — P. 491–502.
32. **Frentzos E., Gratsias K., Theodoridis Y.** Index-based most similar trajectory search // 23rd International Conference on Data Engineering (ICDE). — IEEE Computer Society. Istanbul, 2007. — P. 816–825.
33. **Morse M. D., Patel J. M.** An efficient and accurate method for evaluating time series similarity // Proceedings of the 2007 SIGMOD International Conference on Management of Data. — ACM. Beijing, 2007. — P. 569–580.
34. **Chen Y. [et al.]**. SpADE: On Shape-based Pattern Detection in Streaming Time Series // 23rd International Conference on Data Engineering (ICDE). — IEEE Computer Society. Istanbul, 2007. — P. 786–795.
35. **Keogh E., Pazzani M.** Scaling up Dynamic Time Warping to Massive Datasets // European Conference on Principles of Data Mining and Knowledge Discovery. — Springer. Prague, 1999. — P. 1–11.
36. **Salvador S., Chan P.** FastDTW: Toward Accurate Dynamic Time Warping in Linear Time and Space // 3rd International Workshop on Mining Temporal and Sequential Data. — 2004. — P. 11.
37. **Aach J., Church G.** Aligning gene expression time series with time warping algorithms // Bioinformatics. — 2001. — Vol. 17, no. 6. — P. 495–508.

38. **Yi B., Faloutsos C.** Fast time sequence indexing for arbitrary L_p norms // Proceedings of the 26th International Conference on Very Large Data Bases. — 2000. — P. 385–394.
39. **Scheuermann P.** [et al.]. Querying and mining of time series data: experimental comparison of representations and distance measures // Proceedings of the VLDB Endowment. — 2008. — Vol. 1, no. 2. — P. 1542–1552.
40. **Alia B., Masmoudib Y., Dhouibc S.** DTW-Global Constraint Learning Using Tabu Search Algorithm // Procedia Computer Science. — 2016. — Vol. 82. — P. 12–19.
41. **Rakthanmanon T.** [et al.]. Searching and mining trillions of time series subsequences under dynamic time warping // Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. — 2012. — P. 262–270.
42. **Marteau P.-F., Gibet S.** On Recursive Edit Distance Kernels With Application to Time Series Classification // Transactions on Neural Networks and Learning Systems. — 2014. — Vol. 26, no. 6. — P. 1121–1133.
43. **Haussler D.** Convolution Kernels on Discrete Structures // Technical Report. — Department of Computer Science, University of California, 1999. — Vol. 646. — P. 1–38.
44. **Scholkopf B., Smola A., Muller K.-R.** Nonlinear Component Analysis as a Kernel Eigenvalue Problem // Neural Computation. — 1998. — Vol. 10, no. 5. — P. 1299–1319.
45. **Cuturi M.** [et al.]. A Kernel for Time Series Based on Global Alignments // International Conference on Acoustics, Speech and Signal Processing. — 2007. — Vol. 2. — P. 413–416.
46. **Petitjean F.** [et al.]. Dynamic Time Warping Averaging of Time Series Allows Faster and More Accurate Classification // IEEE International Conference on Data Engineering. — Chicago: IEEE Computer Society, 2014. — P. 470–479.
47. **Alon J., Athitsos V., Sclaroff S.** Online and Offline Character Recognition Using Alignment to Prototypes // 8th International Conference on Document Analysis and Recognition (ICDAR). — IEEE, 2005. — P. 839–843.
48. **Hippisley-Cox J., Vinogradova Y., Coupland C.** Time Series Analysis for selected clinical indicators from the Quality and Outcomes Framework 2001-2006 // QRESEARCH and The Information Centre for Health and Social Care. — 2007.
49. **Palaniappan R.** Biological Signal Analysis. — 1st ed. — London, United Kingdom: Ventus Publishing ApS, 2010. — 137 p.
50. **Sanei S., Chambers J. A.** EEG signal processing. — New York, USA: John Wiley & Sons, 2007. — 313 p.

51. **Narayanan R. M., Smith S., Gallagher K. A.** A Multifrequency Radar System for Detecting Humans and Characterizing Human Activities for Short-Range Through-Wall and Long-Range Foliage Penetration Applications // International Journal of Microwave Science and Technology. — 2014. — Vol. 2014. — P. 1–21.
52. **Kilic D., Ugur O.** Multiresolution analysis of S&P500 time series // Annals of Operations Research. — 2018. — Vol. 260, no. 1. — P. 197–216.
53. **Cortez P.** [et al.]. Multi-scale Internet traffic forecasting using neural networks and time series methods // Expert Systems. — 2012. — Vol. 29, no. 2. — P. 143–155.
54. **Boor C. de.** A Practical Guide to Splines. — Revised edition. — New York, USA: Springer-Verlag., 1978. — 341 p.
55. Data from accelerometer. — URL: http://sourceforge.net/p/mlalgorithms/TSLearning/data/preprocessed%5C_large.csv (дата обр. 19.11.2015).
56. **McKinley P., Levine M.** Cubic Spline Interpolation // College of the Redwoods. — 1998. — Vol. 45, no. 1. — P. 1049–1060.
57. **Garcia-Gonzalez D.** [et al.]. A public domain dataset for real-life human activity recognition using smartphone sensor // Sensors. — 2020. — Vol. 20, no. 8. — URL: <https://www.mdpi.com/1424-8220/20/8/2200>.
58. **Guna J., Humar I., Pogačnik M.** Intuitive gesture based user identification system // 35th International Conference on Telecommunications and Signal Processing (TSP). — IEEE, 2012. — P. 629–633.
59. Electric devices dataset. — URL: <http://www.timeseriesclassification.com/description.php?Dataset=ElectricDevices> (дата обр. 25.07.2020).
60. Dataset listing. — URL: <http://www.timeseriesclassification.com/dataset.php> (дата обр. 27.05.2021).
61. **Villar J. R.** [et al.]. Generalized models for the classification of abnormal movements in daily life and its applicability to epilepsy convulsion recognition // International journal of neural systems. — 2016. — Vol. 26, no. 6.
62. **Williams B. H., Toussaint M., Storkey A. J.** Extracting motion primitives from natural handwriting data // International Conference on Artificial Neural Networks. — Berlin, Heidelberg: Springer, 2006. — P. 634–643.
63. **Petitjean F., Ketterlin A., Gancarski P.** A global averaging method for dynamic time warping, with applications to clustering // Pattern Recognition. — 2011. — Vol. 44, no. 3. — P. 678–693.
64. **Efron B., Tibshirani R.** Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy // Statistical science. — 1986. — Vol. 1, no. 1. — P. 54–75.