

Динамическое выравнивание многомерных временных рядов*

Гончаров А. В., Моргачев Г. И., Смирнов В., Липницкая Т.

morgachev.gi@phystech.edu, smirnov.vs@phystech.edu, tanya.lipnizky@yandex.ru

МФТИ

В работе решается задача выбора оптимальной функции расстояния между двумя наборами временных рядов с общей осью времени. Метод динамического выравнивания одномерных временных рядов использует в качестве функции расстояния стоимость выравнивающего пути. Данный путь получается в результате динамического выравнивания временных осей друг относительно друга. Определенная таким образом функция расстояния называется *функцией стоимости* выравнивания наборов рядов. Предлагается обобщение алгоритма выравнивания на случай наборов, состоящих более чем из одного ряда.

В работе сравниваются функции стоимости, полученные на основе выбора различных функций расстояния между значениями наборов. Анализ полученных функций стоимости проведен на примере решения задач метрической кластеризации и метрического поиска эталонной подпоследовательности для многомерных наборов рядов. Работа также предлагает модификацию оптимизированного алгоритма вычисления функции стоимости выравнивания рядов, позволяющую проводить её вычисления для наборов. Результаты моделирования представлены на основе размеченных данных о деятельности человека, полученных с носимого устройства, а также данных о движении руки в процессе написания текста, полученных с графического планшета.

Обоснованность метрического подхода к решению задачи демонстрируется путем сравнения с результатами на основе авторегрессионной модели.

Ключевые слова: *временные ряды, многомерные временные ряды, DTW.*

1 Введение

Для вычисления функции расстояния между временными рядами применяются различные подходы, одним из которых является поточечный подход: определение расстояния между как суммы расстояний между соответственными точками рядов, слабо отражает сходства объектов в связи с возможными деформациями оси времени: сдвигами и растяжениями [1]. Одним из способов решения этой проблемы является выравнивание временных рядов (Dynamic Time Warping) [2] и его модификации [3, 4]. Данный способ позволяет найти оптимальный выравнивающий путь между рядами, деформированными во времени [5].

* Работа выполнена при финансовой поддержке РФФИ, проект № 00-00-00000. Научный руководитель: Гончаров А. В. Задачу поставил: Гончаров А. В. Консультант: Гончаров А. В.

Стоимость выравнивающего пути используется в качестве расстояния между выравниваемыми объектами. Заданная таким образом функция расстояния для однозначности обозначений называется *функцией стоимости* выравнивания рядов.

Данные, имеющие более одного измерения в каждый момент времени представляются в виде *наборов временных рядов* с общей осью времени. Набор временных рядов будет обозначаться $X = \{X^1, \dots, X^l\}$, каждый $X^i \in \mathbb{R}^n$ – временной ряд длины n . Набор временных рядов далее для краткости называется *рядом* и рассматриваться как вектор-функция от дискретного аргумента – времени. Значение этой функции в момент времени t обозначается \mathbf{X}_t и называется *значением ряда*.

Модификации алгоритма выравнивания для случая наборов описано в [6, 7]. В работе [6] предлагается способ выравнивания наборов рядов, основанный на нормализации исходных данных и нахождении векторной нормы. ???не надо-> В [7] упоминается алгоритм, выполняющий выравнивание соответствующих каналов наборов временных рядов независимо друг от друга, однако такой подход не учитывает исходную природу измерений, так как в выравненных рядах одному моменту времени могут соответствовать не одновременных измерения по различным каналам. <-???

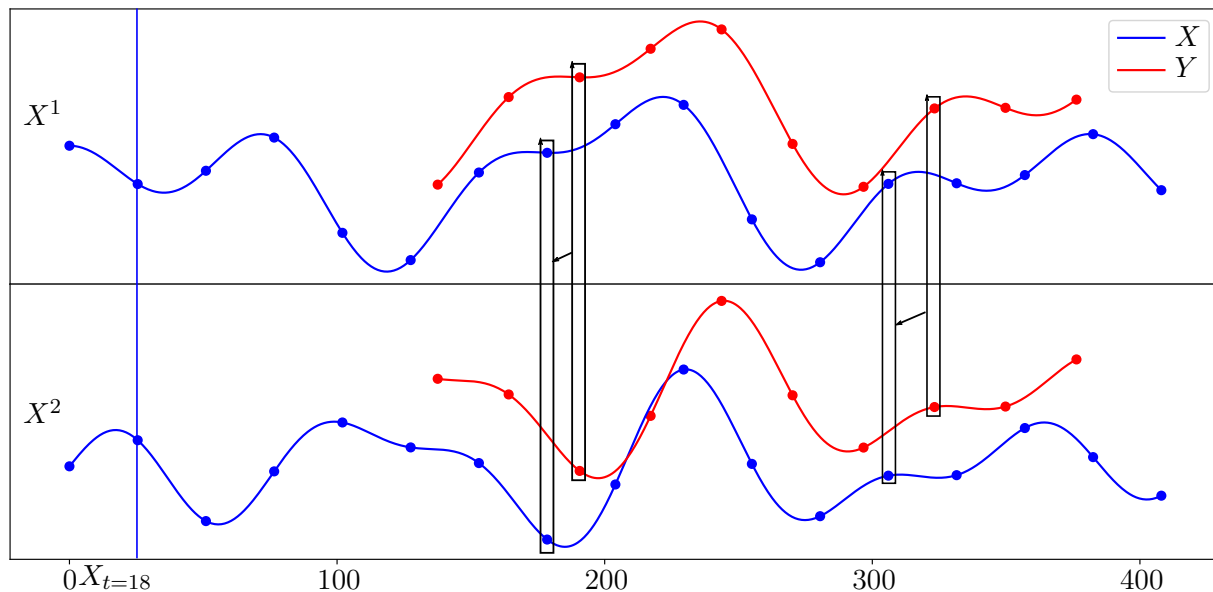


Рис. 1 Иллюстрация выравнивания

При выравнивании рядов [7] вычисляются расстояния между значениями рядов. *Значением ряда* в момент t обозначим $\mathbf{X}_t = [X_t^1, \dots, X_t^l]$, где X_t^i – измерение i -го временного ряда в момент времени t . Таким образом, значение \mathbf{X}_t в момент t принадлежит пространству \mathbb{R}^l . На рис. 1 синей линией обозначено значение набора из двух временных рядов при $t = 18$.

При выравнивании двух рядов относительно общей временной оси, вычисляются расстояния между соответственными значениями. При этом получаемая функция стоимости будет зависеть от выбора функции расстояния между значениями рядов. На рис. 1 стрелками соединены сравниваемые значения, которые совпадут после выравнивания.

В этой работе решается задача выбора оптимальной функции стоимости выравнивания наборов временных рядов с общей осью времени. Сравниваются функции расстояния, порожденные L_1 , L_2 нормами, а также косинусное расстояние.

Качество полученных функций стоимости анализируется в двух задачах, решаемых метрическим методом:

- 1) Поиск шаблонов в наборе временных рядов
- 2) Кластеризация наборов временных рядов

Для каждой из которых введены свои метрики качества $Q_{\text{поиска}}$, $Q_{\text{класт.}}$, соответственно. Задача поиска паттернов ставится как нахождение наиболее близкого участка ряда к заданному паттерну. Задача кластеризации ставится как метрическая иерархическая кластеризация.

В ходе эксперимента, предложенный подход был продемонстрирован на задачах поиска приступов эпилепсии [8] и распознавания рукописных букв [9] для поиска паттернов, а также кластеризации измерений акселерометра телефона в кармане человека при различных видах деятельности [10]. Полученные результаты были сравнены с результатами, полученными на основе использования коэффициентов регрессионных моделей в качестве признаков, описывающих ряд. <---???

2 Постановка задачи выбора функции стоимости

Набором временных рядов называется множество $\{X^1, \dots, X^l\}$, где каждый $X^i \in \mathbb{R}^n$ – временной ряд длины n .

Значением набора из l временных рядов называется вектор $\mathbf{X}_t = [X_t^1, \dots, X_t^l] \in \mathbb{R}^l$, в котором каждому из элементов соответствует значение соответствующего временного ряда в момент времени t , как показано на рис. 1.

Изучается влияния выбора функции расстояния между значениями наборов временных рядов на результат работы DTW на примере двух задач:

- 1) Поиск шаблонов в наборе временных рядов
- 2) Кластеризация наборов временных рядов

Для каждой из которых введены метрики качества $Q_{\text{поиска}}$, $Q_{\text{класт.}}$, соответственно.

Рассматриваются следующие функций расстояния между векторами $\mathbf{X}_t, \mathbf{Y}_\tau$:

- 1) $\rho_{L_1}(\mathbf{X}_t, \mathbf{Y}_\tau) = \|\mathbf{X}_t - \mathbf{Y}_\tau\|_1$;

- 2) $\rho_{L_2}(\mathbf{X}_t, \mathbf{Y}_\tau) = \|\mathbf{X}_t - \mathbf{Y}_\tau\|_2$;
 3) $\rho_{\cosine}(\mathbf{X}_t, \mathbf{Y}_\tau) = 1 - \frac{\mathbf{X}_t^T \mathbf{Y}_\tau}{\|\mathbf{X}_t\|_2 \|\mathbf{Y}_\tau\|_2}$.

Решается задача выбора оптимальной функции расстояния между значениями рядов относительно внешних критериев качества $Q_{\text{поиска}}, Q_{\text{класт.}}$.

$$\rho_i^* = \arg \max_{\rho} Q_i(\rho).$$

Сравнивается качество и скорость работы предложенного подхода с другими используемыми методами:

- 1) определение функции расстояния на основе поточечных расстояний между рядами, при равным длинам рядов:

$$ED(X, Y) = \sum_{i=1}^T \|X_t - Y_t\|_2,$$

- 2) применение авторегрессионной модели для описания ряда:

$$\mathbf{X}_T = \mathbf{c} + \sum_{i=0}^{T-1} a_i \mathbf{X}_{T-i} + \varepsilon_T.$$

3 Алгоритм динамического выравнивания DTW

В качестве метрического расстояния между наборами используется стоимость *пути наименьшей стоимости*.

Задано два набора временных рядов, X, Y длины n . Для вычисления пути требуется построить матрицу размера $n \times n$ с элементами $D_{ij} = \rho(X_i, Y_j)$, где ρ – выбранная функция расстояния, X_i, Y_j – значения рядов. Между наборами рядов строится *выравнивающий путь* W , который минимизирует расстояние между ними.

Определение 1. Путь $W = [w_1, \dots, w_K]$ представляет собой упорядоченное множество пар индексов элементов матрицы D , $w_m = (i, j)_m$, при этом K – длина пути. Кроме того, путь должен удовлетворять следующим условиям.

- **Граничные условия:** $w_1 = (1, 1)$, $w_K = (n, n)$.
- **Непрерывность:** $w_k = (i, j)$, $w_{k-1} = (i', j') : i - i' \leq 1, j - j' \leq 1$. Это ограничение означает, что в пути могут участвовать только соседние элемента матрицы.
- **Монотонность:** $w_k = (i, j)$, $w_{k-1} = (i', j') : i - i' \geq 0, j - j' \geq 0$. Это ограничение гарантирует, что путь не возвращается к пройденному значению.

Определение 2. Стоимостью пути W называется сумма всех соответствующих элементов матрицы:

$$\text{cost}(X, Y, W) = \sum_{(i,j) \in W} D_{ij}.$$

Определение 3. *Оптимальным выравнивающим путем между наборами временных рядов X, Y называется путь минимальной стоимости.*

$$DTW(X, Y) = W^* = \arg \min_W \text{cost}(W)$$

Построение оптимального выравнивающего пути методом DTW проводится с помощью рекуррентной процедуры:

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

Здесь $\gamma(i, j)$ суммарное расстояние, D_{ij} – расстояние в текущей клетке. Элемент γ_{ij} заданной таким образом матрицы γ равен стоимости выравнивающего пути между первыми i и j значениями X и Y соответственно.

Дополнительным ограничением, позволяющим значительно ускорить вычисление пути, является использование полосы Сако–Чибя, ограничивающей максимально возможное отклонение от диагонали матрицы D .

4 Задача поиска паттернов

Рассматривается задача поиска заданного паттерна в наборах временных рядов. Имеется набор временных рядов X длины n , содержащий сегменты класса \mathfrak{P} . Класс \mathfrak{P} – в нашем случае, набор временных рядов длины $m \ll n$.

Определение 4. *Под паттерном класса \mathfrak{P} будет подразумевать матожидание ряда из данного класса.*

Известно k представителей класса \mathfrak{P} , не содержащихся в X .

Необходимо найти участки S , соответствующие данному классу.

Обозначим множество начал таких участков как $\mathfrak{T} = \{t_1, \dots, t_j\}$.

Будем считать участок найденным, если его пересечение с предполагаемым участком более 80% длины m . Поиск будет проводиться путем нахождения расстояния с помощью DTW_ρ между фрагментами набора рядов X и шаблоном, полученным из известных экземпляров класса.

Рассматриваемая функция качества:

$$Q_{\text{поиска}}(DTW_\rho) = \frac{\sum_{i=1}^j [t_i - \text{найден}]}{j}.$$

В ходе эксперимента производилось определение паттернов классов двумя разными способами: путем покомпонентного усреднения известных представителей класса, а также

с использованием алгоритма усреднения DBA [11], в процессе которого также производилось выравнивание временных осей наборов рядов из класса.

5 Задача кластеризации

Исходные данные представляют собой наборы временных рядов длины n . Имеется выборка \mathfrak{X} из N элементов. Для каждого элемента выборки X известна метка класса $y \in \mathfrak{Y}$, где \mathfrak{Y} – множество меток классов.

Определим матрицу попарных расстояния:

$$D(\text{DTW}_\rho(\mathfrak{X})) = \|D_{ij}\|, \quad D_{ij} = \text{DTW}_\rho(X, Z), \quad X, Z \in \mathfrak{X}.$$

И модель кластеризации:

$$f : D \rightarrow \mathfrak{S}^N$$

Где \mathfrak{S} – множество меток кластеров. Будем рассматривать следующие функции качества:

$$Q_{\text{поиска1}}(f(D), \mathfrak{X}) = \frac{1}{|S|} \sum_{s \in \mathfrak{S}} \max_y \frac{N_s^y}{N_s}$$

$$Q_{\text{поиска2}}(f(D), \mathfrak{X}) = \frac{1}{|\mathfrak{S}|} \sum_{s \in \mathfrak{S}} \max_y \frac{(N_s^y)^2}{N_s N^y}$$

Здесь:

- N_s – количество элементов в кластере с меткой s .
- N^y – количество элементов в классе y .
- N_s^y – количество элементов класса y в кластере s .

5.1 Описание алгоритма кластеризации

В качестве алгоритма кластеризации используется иерархическая кластеризация, которая базируется на последовательном слиянии ближайших кластеров. Рассматриваются различные функции расстояния между кластерами:

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

6 Оптимизации

Раздел в разработке

В основу реализации легла работа [12], в которой было использовано большое количество

оптимизаций, позволяющих как сократить время вычисления расстояния между рядами, так и, при возможности, вообще не вычислять его. В ходе работы была произведена попытка обобщить данные оптимизации на случай наборов рядов. Ниже приведено их краткое описание.

6.1 Использование квадрата расстояния

В DTW вычисляется производится вычисление квадратного корня, однако, если упустить этот шаг, относительное расстояние не изменится, поскольку обе функции монотонны и вогнуты. Это упрощает вычисления и позволяет сделать модель легкой для понимания. Таким образом, говоря о DTW, мы подразумеваем квадратные аналоги.

6.2 Использование нижней границы

Для ускорения поиска паттернов с использованием DTW используется оценка снизу на стоимость выравнивающего пути. Данный прием позволяет отбрасывать неподходящие последовательности без дорогостоящего вычисления выравнивающего пути.

В эксперименте используется каскадная нижняя граница. Вначале последовательность проходит проверку на требования LB_{Kim} [13], которая использует расстояние между максимальными и минимальными длинами значений наборов рядов.

Однако, для того чтобы сравнить последовательности, они должны быть нормализованы, поэтому значения двух расстояний между максимальными и минимальными точками могут быть ничтожно малы. В случае если последовательность удовлетворила требованиям LB_{Kim} , происходит вторичная проверка LB_{Keogh} , использующей ED.

Также бывает полезным менять роли сравниваемых последовательностей для LB_{Keogh} , от этого будет меняться решение к какой последовательности применяется нижняя граница, причем результаты вычисления этих границ для каждой из последовательностей при их сравнении в общем случае не будут равны. Однако данный метод применяется опционально и только в том случае, если другие нижние границы не проявили себя.

6.3 Использование верхней границы

При вычислении LB_{Keogh} , мы можем заметить, что текущая сумма расстояний между каждой парой значений превысила определённое (наибольшее возможное) значение, в таком случае мы можем прекратить подсчёт, так как дальнейший действия дадут ещё более высокий результат.

Если вся LB_{Keogh} была посчитана и мы обнаружили, что должны вычислить DTW полностью, есть способ отбросить лишние вычислительные затраты на стадии подсчёта DTW. Если постепенно вычислять DTW слева направо от 1 до k и суммировать частичное накопление DTW с вкладом от LB_{Keogh} от $k + 1$ до n . Сумма $DTW(S_{1:k}^i, S_{1:k}^j) +$

$+LB_{Keogh}(S_{k+1:n}^i, S_{k+1:n}^j)$ является нижней границей для $DTW(S_{1:n}^i, S_{1:n}^j)$. Если в какой-то момент такая нижняя граница превысит верхнюю, расчёт прекращается. Кроме того, расходы на расчёт такой границы незначительны.

Во многих случаях становится полезным изменить порядок поиска верхней границы. Очевидно, что разный порядок поиска приносит разное ускорение, более того существует $n!$ вариантов упорядочивания. Для нахождения оптимального варианта, предлагается сортировать индексы основываясь на абсолютных значениях Z -нормализованных наборов рядов.

7 Эксперимент

7.1 Нахождение подпоследовательностей

Использовались две выборки, первая из которых состояла из временных рядов длиной в 182 значения с тремя каналами, соответствующих координатам X, Y и силе нажатия на экран планшета. Вторая выборка состояла из временных рядов активности мозга в разных состояниях: эпилепсия, ходьба, бег, просмотр картинок и имеет длину элемента в 206 значений. Второй датасет имел многократно меньший размер, но при этом значительно отличался от первого: периоды повторения паттернов значительно меньше.

В обеих выборках из каждого класса выбиралось случайное подмножество представителей, из которых, путем усреднения методами DBA и простым средним, получались поисковые шаблоны. Путем склейки оставшихся рядов в случайном порядке, создавался длинный ряд, в котором производился поиск k наиболее близких фрагментов для каждого из шаблонов с использованием ED и DTW_ρ с различными ρ в качестве функций расстояния.

Для оценки погрешностей измерений, описанная процедура была проведена с большим количеством (20) выборок, полученными из исходной с помощью бутстрепа.

Можно заметить, что на всех данных использование L_1 метрики дало лучшие результаты в данной задаче. При этом оптимизации позволили сократить время работы в более чем 5 раз на большом датасете (цифры), но незначительно в малом, при этом сохранив результаты.

7.2 Кластеризация

В ходе эксперимента были использованы данные акселерометра мобильного телефона. Они представляли собой временные ряды длиной в 600 значений ускорений по осям X, Y, Z . Из них методом бутстрепа были сгенерированы 20 выборок из 600 рядов по 50 значений в каждом. Каждых из рядов принадлежал к одному из четырёх возможных классов. Данные были равномерно распределены по всем классам.

ρ	average	characters			epi		
		Q	t	$t_{\text{no optim}}$	Q	t	$t_{\text{no optim}}$
L_1	DBA	0.83 ± 0.14	2.85 ± 0.43	11 ± 1	0.53 ± 0.10	26 ± 1	25 ± 1
	mean	0.87 ± 0.12	2.77 ± 0.45	14 ± 1	0.49 ± 0.08	26 ± 1	28 ± 2
L_2	DBA	0.80 ± 0.17	2.24 ± 0.30	13 ± 1	0.55 ± 0.07	23 ± 1	26 ± 2
	mean	0.84 ± 0.13	2.17 ± 0.22	10 ± 1	0.48 ± 0.10	23 ± 1	26 ± 2
cosine_dist	DBA	0.77 ± 0.18	3.42 ± 1.00	16 ± 1	0.35 ± 0.10	29 ± 2	38 ± 2
	mean	0.81 ± 0.15	3.43 ± 0.83	14 ± 1	0.27 ± 0.06	26 ± 2	38 ± 1
ED	DBA	0.08	17.511	17.511	0.172	1.620	1.620
	mean	0.09	17.645	17.645	0.172	1.540	1.540

Таблица 1 Поиск паттернов.

Для DTW , ED вычислялись матрицы попарных расстояний между парами рядов, на основе которых проводилась кластеризация описанными выше методами. В качестве функций расстояния между векторами использовались метрики, порождённые L_1 и L_2 нормами. Кроме того, для каждого из временных рядов была обучена авторегрессионная модель, на основе коэффициентов которых также произведена кластеризация. Так как в процессе получения данных были возможны различные положения телефона в кармане, кластеризация проводилась на 4, 12, 24, 36 и 48 кластеров.

В большинстве случаев лучшие результаты показал метод кластеризации *weighted*. При этом, методы выравнивания с L_1 и L_2 метриками показали примерно равные результаты. Простейшая регрессионная модель позволила достичь лучшего качества во всех экспериментах.

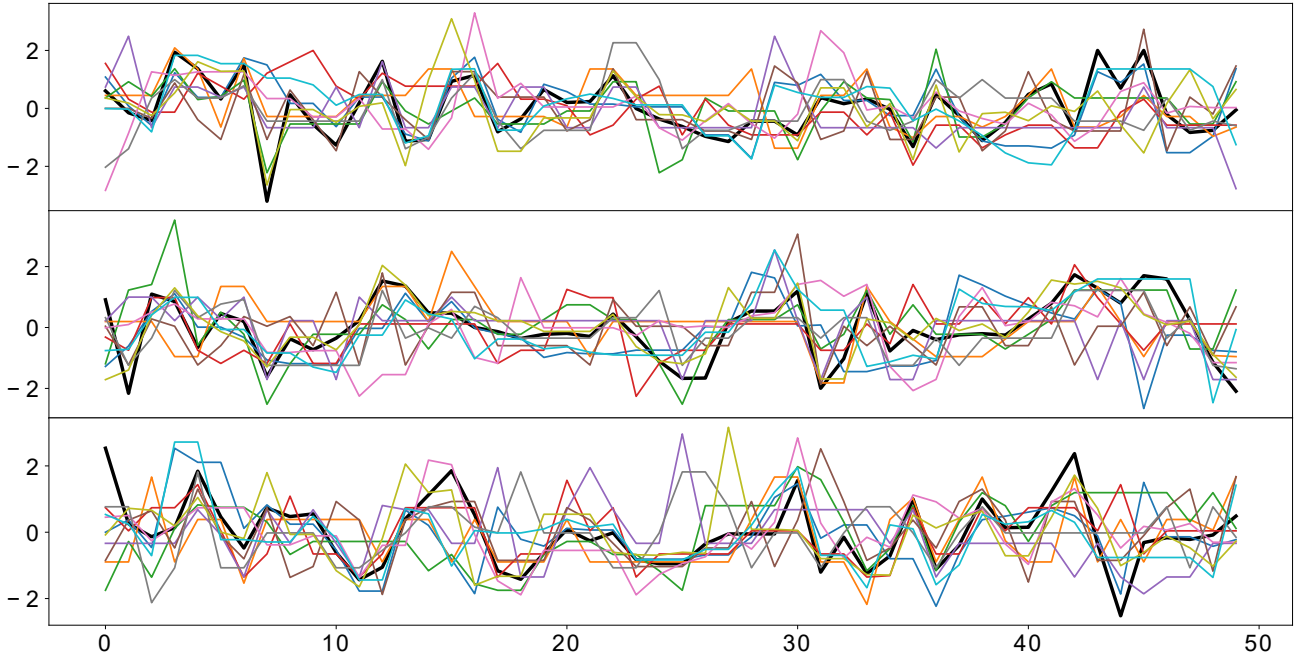
На картинках 2, 3 приводятся примеры рядов, отнесенных к одному кластеру, выравненных относительно случайного набора рядов из данного кластера для L_1 и L_2 метрик соответственно. Кластеризация в данных примерах проводилась с помощью метода *weighted*.

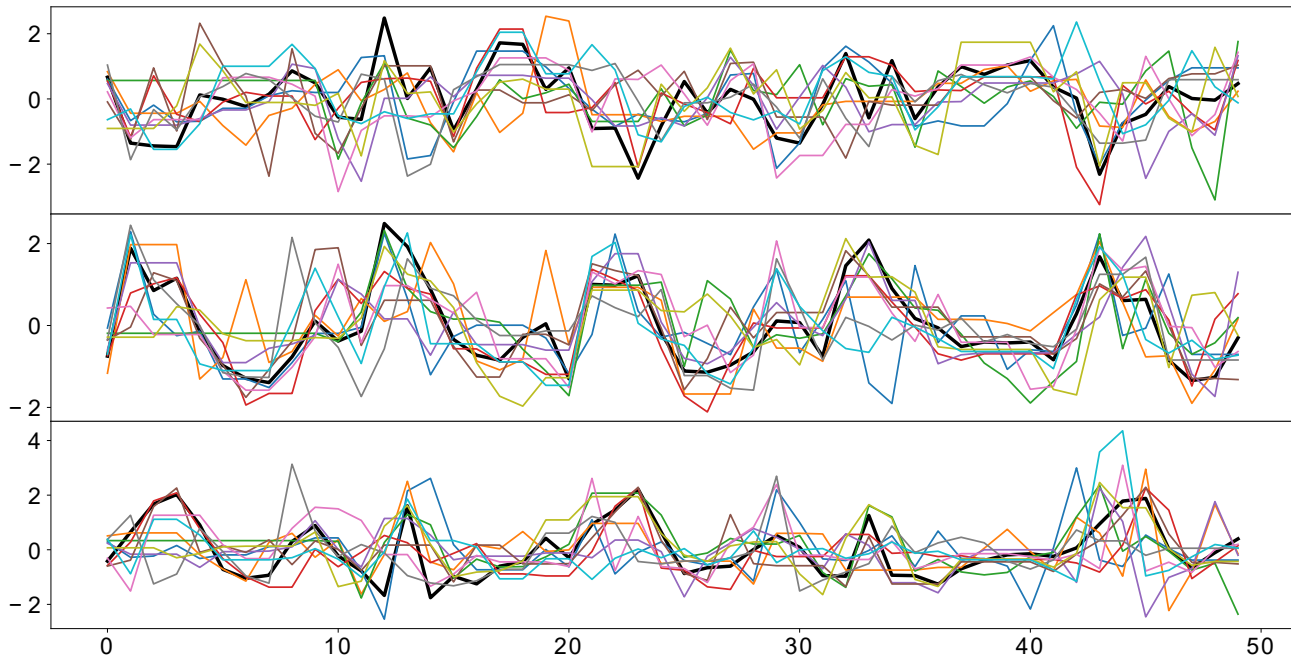
Литература

- [1] Hui Ding, Goce Trajcevski, Peter Scheuermann, Xiaoyue Wang, and Eamonn Keogh. *Querying and mining of time series data: Experimental comparison of representations and distance measures*, volume 1, pages 1542–1552. 2 edition, 8 2008.

ρ	N_{clust}	Q_1			Q_2		
		<i>complete</i>	<i>average</i>	<i>weighted</i>	<i>complete</i>	<i>average</i>	<i>weighted</i>
L_1	4	0.42 ± 0.07	0.51 ± 0.07	0.55 ± 0.06	0.22 ± 0.09	0.31 ± 0.10	0.36 ± 0.09
	12	0.46 ± 0.09	0.54 ± 0.04	0.57 ± 0.04	0.24 ± 0.05	0.33 ± 0.05	0.36 ± 0.05
	24	0.50 ± 0.03	0.57 ± 0.03	0.59 ± 0.03	0.28 ± 0.03	0.36 ± 0.04	0.38 ± 0.04
	36	0.55 ± 0.02	0.60 ± 0.02	0.61 ± 0.03	0.33 ± 0.03	0.40 ± 0.03	0.41 ± 0.05
	48	0.58 ± 0.02	0.63 ± 0.02	0.64 ± 0.03	0.37 ± 0.03	0.43 ± 0.03	0.44 ± 0.04
L_2	4	0.42 ± 0.06	0.47 ± 0.07	0.54 ± 0.06	0.22 ± 0.07	0.26 ± 0.09	0.35 ± 0.09
	12	0.46 ± 0.04	0.53 ± 0.04	0.59 ± 0.04	0.25 ± 0.05	0.32 ± 0.05	0.39 ± 0.05
	24	0.50 ± 0.03	0.57 ± 0.03	0.59 ± 0.03	0.28 ± 0.03	0.35 ± 0.04	0.39 ± 0.03
	36	0.54 ± 0.03	0.60 ± 0.03	0.62 ± 0.03	0.33 ± 0.03	0.40 ± 0.04	0.42 ± 0.03
	48	0.58 ± 0.02	0.63 ± 0.03	0.64 ± 0.03	0.37 ± 0.03	0.43 ± 0.04	0.46 ± 0.04
AR	4	0.52 ± 0.06			0.32 ± 0.08		
	12	0.61 ± 0.05			0.42 ± 0.06		
	24	0.67 ± 0.05			0.51 ± 0.07		
	36	0.70 ± 0.03			0.55 ± 0.05		
	48	0.73 ± 0.03			0.59 ± 0.05		

Таблица 2 Кластеризация.

Рис. 2 Кластеризация с L_1

Рис. 3 Кластеризация с L_2

- [2] Eamonn J. Keogh and Michael J. Pazzani. Scaling up dynamic time warping to massive dataset. In *Proceedings of the Third European Conference on Principles of Data Mining and Knowledge Discovery*, PKDD '99, pages 1–11, London, UK, UK, 1999. Springer-Verlag.
- [3] Stan Salvador and Philip Chan. Toward accurate dynamic time warping in linear time and space. *Intell. Data Anal*, 11(5):561–580, 2007.
- [4] Eamonn J. Keogh and Michael J. Pazzani. Derivative dynamic time warping. In *SIAM International Conference on Data Mining*, 2001.
- [5] Stan Salvador and Philip Chan. Fastdtw: Toward accurate dynamic time warping in linear time and space. In *KDD workshop on mining temporal and sequential data*. Citeseer, 2004.
- [6] G.A. ten Holt, M.J.T. Reinders, and E.A. Hendriks. Multi-dimensional dynamic time warping for gesture recognition. In *Thirteenth annual conference of the Advanced School for Computing and Imaging*, 2007.
- [7] Parinya Sanguansat. Multiple multidimensional sequence alignment using generalized dynamic time warping. 2012.
- [8] Jose Villar, PULA VERGARA, Manuel Menéndez González, Enrique Marín, VÍCTOR M. GONZÁLEZ, and Javier Sedano. Generalized models for the classification of abnormal movements in daily life and its applicability to epilepsy convulsion recognition. *International Journal of Neural Systems*, 26, 04 2016.
- [9] Ben H. Williams, Marc Toussaint, and Amos J. Storkey. Extracting motion primitives from natural handwriting data. In Stefanos Kollias, Andreas Stafylopatis, Włodzisław Duch, and Erkki

- Oja, editors, *Artificial Neural Networks – ICANN 2006*, pages 634–643, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.
- [10] Jennifer R. Kwapisz, Gary M. Weiss, and Samuel A. Moore. Activity recognition using cell phone accelerometers. *SIGKDD Explor. Newsl.*, 12(2):74–82, March 2011.
- [11] François Petitjean, Alain Ketterlin, and Pierre Gançarski. A global averaging method for dynamic time warping, with applications to clustering. *Pattern Recognition*, 44(3):678–693, 2011.
- [12] Thanawin Rakthanmanon, Bilson Campana, Abdullah Mueen, Gustavo Batista, Brandon Westover, Qiang Zhu, Jesin Zakaria, and Eamonn Keogh. Searching and mining trillions of time series subsequences under dynamic time warping. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '12, pages 262–270, New York, NY, USA, 2012. ACM.
- [13] Sang-Wook Kim, Sanghyun Park, and Wesley W. Chu. An index-based approach for similarity search supporting time warping in large sequence databases. *Proceedings - International Conference on Data Engineering*, pages 607–614, 1 2001.