

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 исследуется зависимость качества решения задач метрической кластеризации и метрического поиска эталонной подпоследовательности для многомерных временных рядов от выбранной функции расстояния между объектами. Базовая функция расстояния между временными рядами основана на методе динамического выравнивания оси времени. Однако в базовой реализации алгоритм применим лишь к случаю одномерного временного ряда. Работа предлагает модификацию оптимизированного алгоритма расчета функции расстояния между объектами, основанного на динамическом выравнивании оси времени. Для анализа зависимости качества решения поставленных задач выбраны несколько модификаций базовой функции расстояния. Результаты моделирования представлены в работе на основе размеченных данных о деятельности человека полученных с носимого устройства. 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 описания различных данных широко используются временные ряды. Чтобы найти их сходство вводится функция расстояния, однако стандартный поточечный подход не является информативным вследствие того, что ряды могут содержать общие паттерны, деформированные относительно временной оси: претерпевшие сдвиги либо сжатия [1]. Одним из способов решения этой проблемы является выравнивание временных рядов (DTW) [2] и его модификаций [3]. Этот подход в большом спектре задач позволяет достичь максимального качества среди его аналогов.

В работе рассматривается применения DTW для кластеризации рядов и поиска подпоследовательностей соответствующих некоторому классу событий для случая многомерных рядов. Использование DTW на подобных данных описано в [4, 5]. В работе [4] предлагается способ выравнивания многомерных рядов, основанный на нормализации исходных данных и нахождении векторной нормы. В [5] рассматривается алгоритм, позволяющий выполнить выравнивание временных рядов между координатами. Многомерное DTW предполагает различные варианты выравнивания, такие как выравнивание относительно общей временной шкалы и между соответствующими каналами.

В процессе работы алгоритма DTW происходит вычисление расстояний между точками сравниваемых рядов. Поскольку в многомерном случае координаты точек описываются векторами, на результат будет влиять выбор функций расстояния между ними. Исследование влияние выбора этих функций на качество кластеризации является главной особенностью этой работы. В работе используются функции расстояния порождённые L_1 и L_2 нормами.

Ещё одним стандартным подходом к нахождению сходства между рядами является сравнение представления рядов коэффициентами их регрессионных моделей. Полученная

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

в ходе работы DTW кластеризация сравнивается с кластеризацией на основе авторегрессионной модели.

В статьях [6] [7] рассматриваются различные виды алгоритмов кластеризации временных рядов, среди которых неплохие результаты показывают варианты иерархической кластеризации. Данный вид кластеризации был выбран в качестве базового.

Данные [8] представляют собой измерения акселерометра некоторого носимого устройства, например мобильного телефона, находящегося в кармане человека, и используется для идентификации действия человека в конкретный момент времени. Данные разделены на 6 классов: ходьба, бег, подъём по лестнице, спуск по лестнице, сидение, лежание.

2 Постановка задачи

Временным рядом называется упорядоченная последовательность измерений $S = s_1, s_2, \dots, s_n$, где n - длина временного ряда.

Поскольку мы используем многомерные временные ряды, то $s_i, i \in 1 \dots n$, представляют собой вектор размерности l . Например, при рассмотрении задачи идентификации определенного движения человека вектор является трёхмерным с координатами (x, y, z) .

Имеется множество функций расстояния между векторами \mathbf{R} :

$$\mathbf{R} = \{\rho : \mathbb{R}^l \times \mathbb{R}^l \rightarrow \mathbb{R}_+\}$$

Каждой функции расстояния ρ между векторами соответствует функция расстояния между временными рядами:

$$\text{DTW}_\rho : \mathbf{S} \times \mathbf{S} \rightarrow \mathbb{R}_+.$$

Целью эксперимента является изучения влияния выбора поточечной метрики на результат работы DTW на примере двух задач, описанных в разделе 3:

- 1) Поиск шаблонов в ряду
- 2) Кластеризация рядов

Для данных задач задаются соответствующие метрики качества Q_i . Тогда, решаемая задача определения оптимальной метрики:

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

Рассматриваются следующие функций расстояния между векторами:

- 1) $L_1(v, u) = \sum_{l=1}^l |v_l - u_l|$
- 2) $L_2(v, u) = \sqrt{\sum_{l=1}^l (v_l - u_l)^2}$, евклидово расстояние, также будем обозначать его $ED(u, v)$
- 3) $\text{cosine}(v, u) = 1 - \frac{(v, u)}{\sqrt{(v, v)}\sqrt{(u, u)}}$

Кроме того, производится сравнения качества и скорости работы данного подхода с другими используемыми методами:

- 1) Функция расстояния на основе поточечных расстояний между рядами, при равным длинах рядов:

$$ED(S_1, S_2) = \sum_{i=1}^m \|s_{1i} - s_{2i}\|_2$$

2) Применение авторегрессионной модели для описания ряда.

3 Рассматриваемые задачи

3.1 Поиск паттернов

Рассматривается задача поиска заранее известных паттернов во временном ряду. Имеется временной ряд S длины n , содержащий сегменты класса P . Класс P - в нашем случае, набор временных рядов длины $m \ll n$.

Известно k представителей класса P , не содержащихся в S .

Необходимо найти участки S , соответствующие данному классу.

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

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 ряда S и шаблоном, полученным из известных экземпляров класса.

Рассматриваемая функция качества:

$$Q_3(DTW_\rho) = \frac{\sum_{i=1}^j [t_i - \text{найден}]}{j}.$$

3.2 Кластеризация

Исходные данные представляют собой временные ряды (длиной n). Имеется выборка $S \subset \mathbb{S}$ из N элементов. Для каждого элемента выборки s_i известна метка класса $y_i \in \mathbb{Y}$, где \mathbb{Y} - множество меток классов.

Определим матрицу попарных расстояния:

$$D(DTW_\rho(S)) = \|D_{ij}\|, \quad D_{ij} = DTW_\rho(S_i, S_j), \quad S_i, S_j \in S.$$

И модель кластеризации:

$$f : D \rightarrow Z^N$$

Где Z - множество меток кластеров. Будем рассматривать следующие функции качества:

$$Q_1(f(D), S) = \frac{1}{|Z|} \sum_{z \in Z} \max_y \frac{N_z^y}{N_z}$$

$$Q_2(f(D), S) = \frac{1}{|Z|} \sum_{z \in Z} \max_y \frac{(N_z^y)^2}{N_z N^y}$$

Здесь:

- N_z - количество элементов в кластере с меткой z .
- N^y - количество элементов в классе y .
- N_z^y - количество элементов класса y в классе z .

4 Описание основных методов

Для построения функции выравнивания и проверки её качества используются модель DTW (и её оптимизация).

4.1 Описание функции расстояния между объектами

В данной работе в качестве метрического расстояния между объектами предлагается использовать строимость *пути наименьшей стоимости* между объектами.

Dynamic time warping - измерение расстояния между двумя временными рядами.

Задано два временных ряда, X длины m_1 и Y длины m_2 .

$$X = x_1, x_2, \dots, x_i, \dots, x_{m_1}$$

$$Y = y_1, y_2, \dots, y_j, \dots, y_{m_2}$$

$$x_i, y_j \in \mathbb{R}^n$$

Требуется построить матрицу размера $m_1 \times m_2$ с элементами $D_{ij} = d(x_i, y_j)$, где d - выбранная метрика.

Чтобы найти наибольшее соответствие между рядами нужно найти выравнивающий путь W , который минимизирует расстояние между ними. W - набор смежных элементов матрицы D , $w_k = (i, j)_k$.

$$W = w_1, w_2, \dots, w_k, \dots, w_K$$

$\max(m_1, m_2) \leq K \leq m_1 + m_2 + 1$, где K -длина выравнивающего пути

Выравнивающий путь должен удовлетворять следующим условиям:

- 1) $w_1 = (1, 1)$, $w_K = (m_1, m_2)$
- 2) $w_k = (a, b)$, $w_{k-1} = (a', b') : a - a' \leq 1, b - b' \leq 1$
- 3) $w_k = (a, b)$, $w_{k-1} = (a', b') : a - a' \geq 0, b - b' \geq 0$

Оптимальный выравнивающий путь должен минимизировать выравнивающую стоимость пути:

$$DTW(X, Y) = \sum_{k=1}^K w_k$$

Путь находится рекуррентно:

$\gamma(i, j) = d(q_i, c_j) + \min(\gamma(i-1, j-1), \gamma(i-1, j), \gamma(i, j-1))$, где $\gamma(i, j)$ суммарное расстояние, $d(q_i, c_j)$ расстояние в текущей клетке.

Кроме того, выравнивающий путь ограничивают тем, насколько он может отклоняться от диагонали. Типичным ограничением является полоса Сако-Чибэ, накладывающая условие, что путь искривления не может отклоняться от диагонали больше, чем на определённый процент клеток.

4.2 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*: $d(A, B) = \sum_{a \in A, b \in B} \frac{d(a, b)}{(|A| * |B|)}$

5 Оптимизации

В основу нашей реализации легла работа [9], 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 мы постарались обобщить эти оптимизации на многомерный случай. Ниже приведено их краткое описание.

5.1 Использование квадрата расстояния

В DTW вычисляется квадратный корень, 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.

5.2 Использование нижней границы

Для того чтобы ускорить последовательный поиск в DTW, используется нижняя граница (Lower Bounding), чтобы отбросить неподходящие последовательности (подпоследовательности). Оптимизация ускоряет поиск ещё и потому что не требует затратных вычислений.

В эксперименте используется каскадная нижняя граница. Вначале последовательность проходит проверку на требования LB_{Kim} , которая использует расстояние между максимальными значениями рядов и минимальными значениями 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.

5.3 Использование верхней границы

При вычислении LB_{Keogh} , мы можем заметить, что текущая сумма расстояний между каждой парой точек привнесла определённое (наибольшее возможное) значение, в таком случае мы можем прекратить подсчёт, тк 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. Чтобы найти оптимальный вариант, предлагается отсортировать индексы основанных на абсолютных значениях Z-нормализованной последовательности. Одно значение временного ряда S_i сравнивается со многими из ряда S_j , которые далее сортируются по убыванию вклада ED.

6 Эксперимент

6.1 Нахождение подпоследовательностей

Использовались две выборки, первая из которых состояла из временных рядов длиной в 182 точки 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 точек. Второй датасет имел многократно меньший размер, но при этом он значительно отличается 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.

ρ	average	characters			epi		
		Q	t	$t_{no\ optim}$	Q	t	$t_{no\ optim}$
L_1	DBA	0.866	2.117	10.064	0.744	14.335	13.064
	mean	0.901	2.524	10.819	0.744	13.541	13.912
L_2	DBA	0.831	1.308	9.628	0.687	12.342	13.205
	mean	0.864	1.255	10.495	0.687	14.199	12.738
cosine_dist	DBA	0.805	3.221	13.650	0.687	12.342	13.205
	mean	0.819	3.142	14.285	0.687	14.199	12.738
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 Эксперимент 1: поиск паттернов

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.

6.2 Кластеризация

В ходе эксперимента были использованы данные акселерометра мобильного телефона. Они представляли собой временные ряды длиной в 600 точек ускорений по осям X, Y, Z . Из них была сгенерирована выборка из 2048 рядов по 50 точек. Каждых из рядов принадлежал к одному из шести возможных классов. Данные были равномерно распределены по всем классам.

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 24, 36 и 48 кластеров.

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	24	0.506	0.585	0.638	0.273	0.376	0.449
	36	0.533	0.620	0.616	0.299	0.425	0.414
	48	0.556	0.639	0.631	0.330	0.443	0.431
L_2	24	0.488	0.622	0.626	0.270	0.417	0.425
	36	0.498	0.646	0.643	0.270	0.455	0.449
	48	0.534	0.648	0.653	0.270	0.455	0.462

Таблица 2 Эксперимент 2: кластеризация

В большинстве случаев лучшие результаты показал метод кластеризации *weighted*. При этом, методы выравнивания с L_1 и L_2 метриками показали примерно равные результаты.

На картинках 1, 2 приводятся примеры рядов, отнесенных к одному кластеру, выравненных относительно случайного ряда из данного кластера для L_1 и L_2 метрик соответственно. Кластеризация в данных примерах проводилась с помощью метода *weighted*.

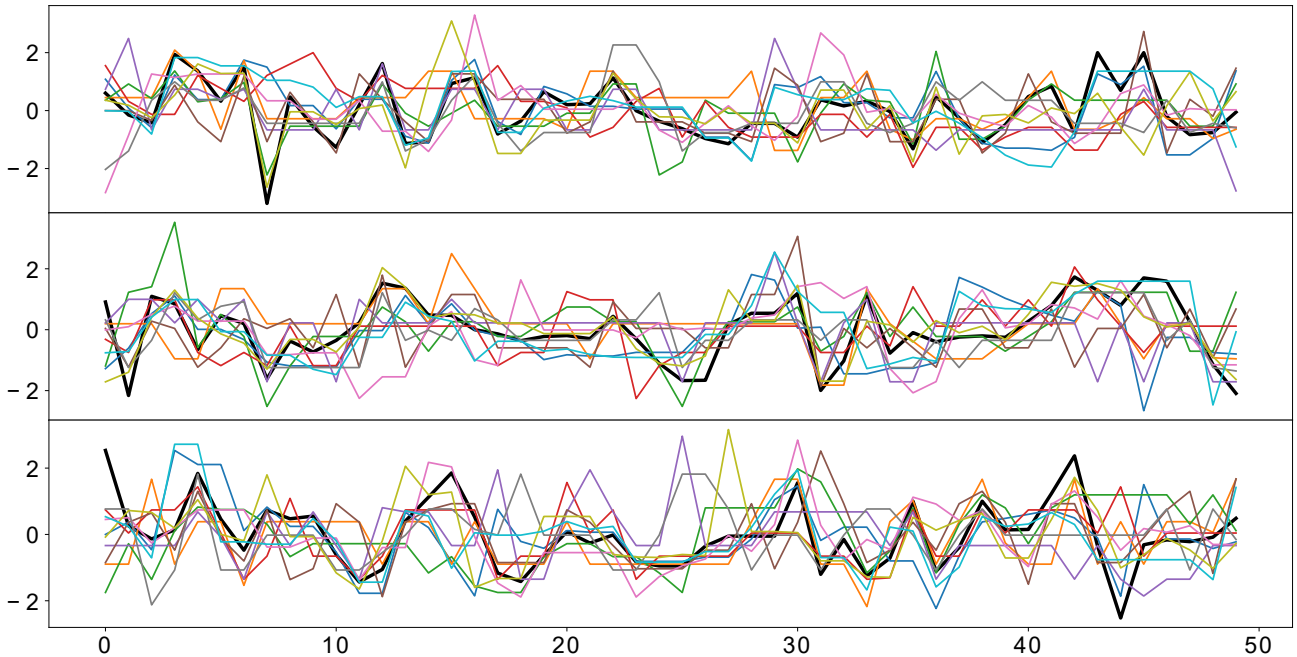
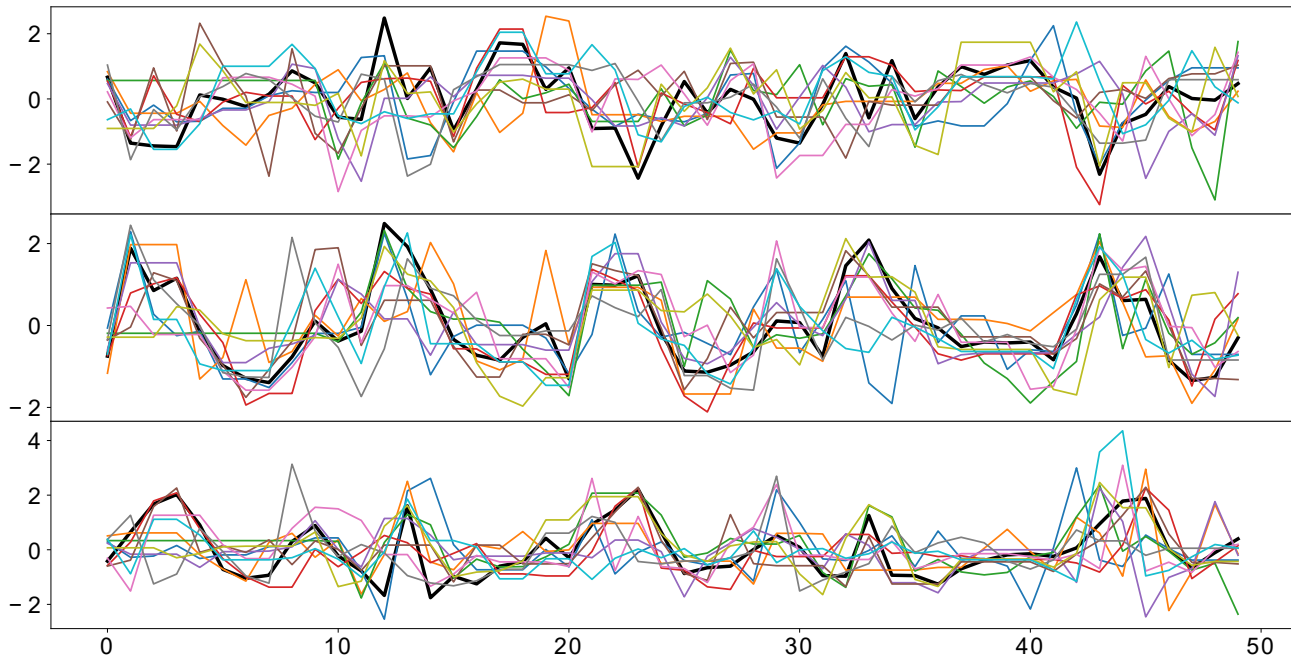


Рис. 1 Кластеризация с L_1

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.

Рис. 2 Кластеризация с L_2

- [2] Eamonn J. Keogh and Michael J. Pazzani. Derivative dynamic time warping. In *SIAM International Conference on Data Mining*, 2001.
- [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] 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.
- [5] Parinya Sanguansat. Multiple multidimensional sequence alignment using generalized dynamic time warping. 2012.
- [6] T. Warren Liao. Clustering of time series data—a survey. *Pattern Recognition*, 38(11):1857 – 1874, 2005.
- [7] Saeed Aghabozorgi, Ali Seyed Shirkhorshidi, and Teh Ying Wah. Time-series clustering – a decade review. *Information Systems*, 53:16 – 38, 2015.
- [8] 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.
- [9] 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.