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 данной работе исследуется кластеризация многомерных временных рядов с использованием алгоритма DTW. При использовании DTW в многомерном случае возникает проблема определения функций расстояния между элементами временных рядов. Основной целью статьи является нахождение зависимости качества кластеризации от выбора этой функции расстояния. В связи с повышением размерности возникает вопрос эффективности и применимости DTW на многомерных рядах. В качестве прикладной задачи исследуется кластеризация размеченных данных о деятельности человека полученных с акселерометра. Оценка качества кластеризации производится при сравнении с результатами кластеризации на основе авторегрессионной модели и анализу распределения классов данных в полученных кластерах.

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 в случае многомерных временных рядов. Использование DTW на подобных данных описано в [4], [5]. В работе [4] предлагается способ выравнивания многомерных рядов, основанный на нормализации исходных данных и нахождениии 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 соответсвующими 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 в ходе работы DTW кластеризация сравнивается кластеризацией на основе авторегрессионной модели.

* 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 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 индентификации действия человека в конкретный момент времени. Данные разделены на 6 классов: ходьба, бег, подъём по лестнице, спуск по лестнице, сидение, лежание.

2 Постановка задачи

Временнным рядом называется упорядоченная последовательность $S_i = s_1, s_2, ..., s_n$, где n - длина временного ряда, $i \in l$, l - количество каналов.

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

Пусть задано множество временных рядов $\mathbb{S} \subset \mathbb{R}^{l \times n}$.

$\forall S_i \in \mathbb{S}$ задано $y_i \in \mathbb{Y}$ - множество меток классов.

Пусть есть множество функций расстояния между векторами R:

$$\mathrm{R} = \{\rho : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^+\}$$

Пусть S_i, $S_j \in \mathbb{S}$, тогда определим некоторые метрики, которые возмём в качестве функций расстояния между фекторами:

1. $L_2 : ED(S_i, S_j) = \sqrt{\sum_{k=1}^{n} (s_{ik} - s_{jk})^2}$

Функция расстояния рассчитывается для каждой координаты вектора s_i, $i \in n$ независимо от отстальных координат.

Исходные данные представляют собой временные ряды (длиной n), однако в первом эксперименте мы будем использовать подпоследовательности временных рядов длины $m \in n$, чтобы выделить в исходном временном ряде определенные области, которые являются общеизвестными паттернами.

Для каждой функции расстояния между векторами существует соответствующая функция расстояния между временными рядами:

$$g_\rho : \mathbb{S} \times \mathbb{S} \to \mathbb{R}^+$$

Возьмем выборку $S \subset \mathbb{S}, \quad |S| = N$

Определим матрицу попарных расстояния:

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

Определим кластеризатор:

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

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.

Тогда, решаемая задача:

$$Q_i(D(g_\rho(S), S) \to \max_\rho$$

3 Описание основных методов

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

3.1 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 n и Y длины m.

$$X = x_1, x_2, ..., x_i, ..., x_n$$
$$Y = y_1, y_2, ..., y_j, ..., y_m$$
$$x_i, y_j \in \mathbb{R}^n$$

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

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

$W = w_1, w_2, ..., w_k, ..., w_K$

$max(n, m) \leqslant K \leqslant m + n + 1$, где K-длина выравнивающего пути

Выравнивающий путь должен удовлетворять следующим условиям:

1. $w_1 = (1, 1)$, $w_K = (n, m)$
2. $w_k = (a, b)$, $w_{k-1} = (a', b') : a - a' \leqslant 1$, $b - b' \leqslant 1$
3. $w_k = (a, b)$, $w_{k-1} = (a', b') : a - a' \geqslant 0$, $b - b' \geqslant 0$

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-Чиба, в которой говорится, что путь искривления не может отклоняться от диагонали больше, чем на определённый процент клеток.

3.2 Оптимизации

3.3 Использование квадрата расстояния

В DTW и ED 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 и ED, мы подразумеваем их квадратные аналоги.

3.4 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.

3.5 Использование верхней границы

При вычислении ED или LB_{Keogh}, мы можем заметить, что текущая сумма расстояний между каждой парой точек привысила 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 $DTW(S^i_{1:k}, S^j_{1:k}) + + LB_{Keogh}(S^i_{k+1:n}, S^j_{k+1:n})$ является нижней границей для $DTW(S^i_{1:n}, S^j_{1:n})$. Если в какой-то момент такая нижняя граница превысит верхнюю, расчёт прекращается. Кроме того, расходы на расчёт такой границы незначительны.

Практически очевидно, что для сравнения двух временных рядов они должны быть нормализованы. Однако она занимает больше времени, чем подсчёт ED. Таким образом было решено объединить ED (LB_Keogh) с Z-нормализацией. Постепенно вычисляя нормализацию, мы в той же точке вычисляем ED (LB_{Keogh}). Это позволяет отбросить неподходящую последовательность не только на стадии подсчёта дистанции, но и на стадии нормализации.

Во многих случаях становится полезным изменить порядок поиска верхней границы. Очевидно, что разный порядок поиска приносит разное ускорение, более того существует n! выриантов упорядочивания. Чтобы найти оптимальный вариант, предлагается отсортировать индексы основанных на абсолютных значениях Z-нормализованной последователь-

ности. Одно значение временного ряда S_i сравнивается со многими из ряда S_j, которые далее сортируются по убыванию вклада ED. Такая сортировка может быть применена как ED и LB_{Keogh}, так и к отбрасыванию неподходящей последовательности на стадии нормализации.

3.6 Использование многоядерных процессоров

Стоит отметить, что при правильном использовании можно добиться практически линейного ускорения с помощью многоядерных процессоров. Однако оптимизации методов полностью затмевают эти улучшения.

3.7 План эксперимента

Эксперимент № 1

1. глеб достал паттерны датасет соответствующий эпилепсии, склеил, добавил данные между ними

2. Использовал алгоритм чтобы выделить паттерны среди данных из приступа эпилепсии прогнав многомерный массив с данными паттернов

3. смотрим насколько успешно, кластеризуем и смотрим по метрике качетва

4. Исследуем разные метрики дистанции (ищем лучшую)

Эксперимент №2

1. сравниваем пул рядов между собой и составляем матрицу их попарных расстояний чтобы кластеризовать на основе алгоритма из 1 эксперимента (оптимизации бесполезны не изем лучшую сравниваем одинаковую по длине, best so far просто может откидывать неподходящие)

2. кластеризуем и смотрим метрикик дистанции и качества

4 Заключение

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.

[2] Eamonn J. Keogh and Michael J. Pazzani. Derivative dynamic time warping. In *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.