

Matrix time warping

Временные ряды представляют собой набор измерений, упорядоченных по оси времени. Во многих задачах: классификации или поиске оптимальной подпоследовательности, метрические методы анализа дают высокое качество решения задачи. Но из-за множества сторонних, случайных факторов временные ряды относительно оси времени могут быть деформированы нелинейно, что накладывает существенные ограничения на применимые методы определения расстояния. Динамическое выравнивание временных рядов позволяет избежать указанной проблемы и выравнивать два объекта оптимальным образом для подсчета расстояния между ними. В настоящей работе предлагается ввести обобщение метода динамического выравнивания для случаев более сложных объектов, которые определены на двух осях времени. В частном, дискретном случае такие объекты представимы в виде матриц, а метод динамического выравнивания временных рядов переходит в метод динамического выравнивания матриц. Определена как функция расстояния, устойчивая к нелинейным деформациям объекта относительно двух временных шкал, так и выравнивающий путь между объектами. Рассмотрены свойства созданной функции расстояния, а для оценки метода на практике проведен вычислительный эксперимент. Решается задача метрической классификации объектов: модельных данных и данных из датасета MNIST.

Key words: функция расстояния, динамическое выравнивание, расстояние между матрицами, нелинейные деформации времени

1 Введение

Временные ряды являются объектами сложной структуры, которые представляют собой упорядоченное множество измерений физической величины, а эти значения возможно получать в различные моменты времени. Возможность собирать данные в реальном времени расширила сферы деятельности, где появляются объекты похожей структуры: данные о продажах, погоде, котировках, биомедицинских и биометрических измерениях и т.д. (ссылки) При этом точность измерения подобных значений может существенно варьироваться, а отрезки времени — отличаться между измерениями. В большинстве прикладных задач в собранных данных встречаются нелинейные деформации оси времени: глобальные и локальные сдвиги, растяжения и сжатия (ссылка).

Вследствие наличия огромного числа разнородных данных, а также актуальности решения задач анализа временных рядов: классификации, детектирования паттернов, кластеризации и т.д., в литературе появилось большое число методов обработки вре-

менных рядов. (ссылка) Однако возможно выделить основные группы методов, с которыми можно столкнуться в литературе. Первая группа методов основана на анализе коэффициентов различных моделей, с помощью которых временные ряды могут быть описаны. (ссылка) Еще одна группа образуется вокруг создания признакового описания таких сложных объектов (ссылка) Третью группу возможно описать как основанную на анализе формы (ссылка) И, наконец, выделяются методы, которые получены при помощи комбинации описанных выше подходов (ссылка).

Задачи кластеризации и поиска оптимальной или фиксированной подпоследовательности в данных не могут быть решены без использования метрических методов анализа временных рядов. Также как и задача классификации при отсутствии выборки достаточного для решения объема. Метрические методы позволяют находить схожие объекты внутри набора данных. Существует множество подходов к определению функции расстояния над временными рядами: расстояние Хаусдорфа (ссылка), MODH (ссылка), расстояние, основанное на НММ (ссылка), Евклидово расстояние в исходном пространстве или в пространстве сниженной размерности (ссылка), LCSS и многие другие. В литературе неоднократно показано, что в случае локальных или глобальных деформаций времени при решении задач, требующих анализа исходной формы временного ряда метод динамического выравнивания оси времени DTW превосходит другие функции расстояния по качеству итогового решения задачи, так как при наличии смещений двух объектов друг относительно друга приходится выравнивать их оптимальным образом, для вычисления расстояния между ними. (ссылка)

В работе предлагается перейти от рассмотрения объекта вида $s(t)$, задающих временные ряды, к более общему случаю $s(t)$, в частности, рассмотреть объекты, которые будут определены на двух осях времени $s(t_1, t_2)$. Подобное обобщение понятия временного ряда возможно представить в качестве математического объекта. Однако в природе представить такой объект оказывается затруднительно.

В случае дискретной и ограниченной сверху шкалы времени временной ряд представляет собой вектор фиксированной размерности. По аналогии с этим, описанный объект будет представлен матрицей. С одной стороны, любое изображение может быть представлено в виде матрицы. С другой стороны, для такого объекта не будут выполнены основные гипотезы порождения данных, а именно: два схожих объекта не могут быть получены путем нелинейно деформации осей времени. Приближенными к описанным свойствами обладает частотный спектр сигнала, где одна ось определяет время, а другая — частоту, величину, обратную времени.

Между двумя подобными объектами, матрицами, в случае локальных флуктуаций затруднительно определить устойчивую к деформациям осей времени функцию рассто-

яния. Работа исследует возможность определения такой функции между объектами на основе принципов построения метода динамического выравнивания временных рядов DTW.

В задачах компьютерного зрения решаются задачи поиска схожих элементов на паре изображений. Для этого используются методы, позволяющие построить локальное описание выделенного пикселя изображения, а затем найти ближайший к нему на другом. (ссылка) Путем применения подобных преобразований ко множеству пикселей изображения возможно получить отношение между пикселями, а далее воспользоваться Евклидовым расстоянием для вычисления функции расстояния между изображениями.

Целью данной работы является построение proof-of-the-concept для метода, основанного на динамическом выравнивании осей времени для объектов сложной структуры. Метод динамического выравнивания временных рядов определен только для объектов с одним измерением времени. Однако концепции, используемые на каждой стадии вычисления оптимального выравнивания, могут быть обобщены на более сложный случай. Работа исследует свойства построенного метода и проводит сравнение результатов применения метода к задачам классификации изображений с результатами использования функции расстояния L_2 .

Для получения результатов предложенной моделью решается задача метрической классификации над объектами-матрицами низкой размерности, так как вычислительная сложность предлагаемого метода высока. Используются два типа данных: модельные данные, которые согласуются с выдвинутой гипотезой порождения данных для временных рядов, а также подмножество датасета MNIST сниженной размерности.

2 Постановка задачи

В работе поставлена задача поиска функции расстояния между объектами сложной структуры. Функция расстояния должна быть устойчива к глобальным и локальным линейным и нелинейным деформациям временной шкалы. Ниже приведены две постановки задачи, с помощью которых можно определить свойства какой-либо функции расстояния, оценить ее качество и провести сравнение нескольких вариантов.

Первая постановка задачи касается общего свойства функций расстояния: “объединение” схожих объектов и “разделение” непохожих объектов.

Вторая постановка задачи близка к первой и заключается в проведении метрической классификации методом ближайшего соседа. В литературе при классификации времен-

ных рядов такой способ задает базовое решение высокой точности в случае правильного выбора функции расстояния.

Постановка задачи поиска функции расстояния между двумя объектами:

Пусть на двух временных осях заданы объекты вида $\mathbf{A}(t_1, t_2) \in \mathbb{R}^{n_1 \times n_2}$. Пусть функция $G_w(\mathbf{A}) : \mathbb{R}^{n_1 \times n_2} \rightarrow \mathbb{R}^{\hat{n}_1 \times \hat{n}_2}$ задает все допустимые трансформации исходного объекта \mathbf{A} : глобальные сдвиги, локальные линейные и нелинейные деформации, а параметр w функции G фиксирует вид этих трансформаций.

Будем считать, что объект $\mathbf{B} \in \mathbb{R}^{n_1 \times n_2}$ может быть получен из объекта \mathbf{A} при помощи допустимых трансформаций, если $\exists \hat{w} : \mathbf{B} = G_{\hat{w}}(\mathbf{A})$.

Качество функции расстояния между двумя объектами обозначенной структуры $\rho : \mathbb{R}^{\hat{n}_1 \times \hat{n}_2} \times \mathbb{R}^{\hat{n}_3 \times \hat{n}_4} \rightarrow \mathbb{R}^+$ будем оценивать на выборке $\mathfrak{D} = \{\mathbf{A}_i\}_{i=1}^m$ объектов вида $\mathbf{A}_i \in \mathbb{R}^{n \times n}$.

Предположим, что существуют допустимые трансформации двух объектов $\mathbf{A}, \mathbf{B} \in \mathfrak{D}$, задающиеся параметрами w_1 и w_2 , такие, что $G_{w_1}(\mathbf{A}) = G_{w_2}(\mathbf{B})$. Также предположим, что множество объектов $\{\mathbf{C}_i\}$ задает множество объектов, для которых выполняется: $\nexists w_i, w : G_w(\mathbf{A}) = G_{w_i}(\mathbf{C}_i)$.

Задача работы — поиск функции расстояния ρ , значение которой между \mathbf{A} и \mathbf{B} без ограничения общности меньше, чем между объектом \mathbf{A} и любым объектом \mathbf{C}_i : $\forall i \quad \rho(\mathbf{A}, \mathbf{B}) < \rho(\mathbf{A}, \mathbf{C}_i)$.

Функцией ошибки для функции расстояния ρ на выборке \mathfrak{D} примем долю объектов, для которого указанное неравенство не выполняется:

$$S_\rho(\mathfrak{D}) = \frac{1}{m} \sum_i \prod_{\{\mathbf{B}_j\}_i} \prod_{\{\mathbf{C}_k\}_i} [\rho(\mathbf{A}_i, \mathbf{B}_j) < \rho(\mathbf{A}_i, \mathbf{C}_k)], \text{ где}$$

$$\{\mathbf{B}_j\}_i = \{\mathbf{B} \in \mathfrak{D} | \exists w_1, w_2 : G_{w_1}(\mathbf{A}_i) = G_{w_2}(\mathbf{B})\},$$

$$\{\mathbf{C}_k\}_i = \{\mathbf{C} \in \mathfrak{D} | \nexists w_1, w_2 : G_{w_1}(\mathbf{A}_i) = G_{w_2}(\mathbf{C})\}.$$

Постановка задачи выбора функции расстояния ρ сводится к задаче минимизации функции ошибки:

$$\rho = \underset{\rho}{\operatorname{argmin}} S_\rho(\mathfrak{D}).$$

Прикладное использование функции расстояния:

Задана выборка $\mathfrak{D} = \{(\mathbf{A}_i, y_i)\}_{i=1}^m$, состоящая из пар объект — ответ. Объектами служат объекты сложной структуры: $\mathbf{A}_i \in \mathbb{R}^{n \times n}$, а ответами являются метки класса —

$y_i \in Y = \{1, \dots, E\}$, где $E \ll m$. Выборка разделена на обучение $\mathfrak{D}_l = \{(\mathbf{A}_i, y_i)\}_{i=1}^{m_1}$ и контроль $\mathfrak{D}_{m_1}^{m_1+m_2}$.

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 :

$$\hat{y} = f(\mathbf{B}|\rho) = y_{\underset{i=1, \dots, m_1}{\operatorname{argmin}} \rho(\mathbf{B}, \mathbf{A}_i)}$$

Функция ошибки S модели f для задачи классификации — доля ошибок на контрольной выборке:

$$S(f|\rho) = \frac{1}{m_2} \sum_{i=1}^{m_1} [f(\mathbf{A}_i|\rho) \neq y_i].$$

Требуется выбрать функцию расстояния ρ для модели классификации $f : \mathbb{R}^{n \times n} \rightarrow Y$, минимизирующую функцию ошибки S на контрольной выборке:

$$f = \underset{\rho}{\operatorname{argmin}} (S(f|\rho)). \quad (1)$$

3 MatrixDTW определение

В качестве оптимальной функции расстояния в случае допустимых трансформаций над объектами указанной структуры предлагается использовать функцию расстояния Dynamic Time Warping, модифицированную для случая выравнивания двойной шкалы времени. Ниже приведены необходимые определения и мотивация сравнения со случаем одной шкалы времени.

Определение 1: Даны два объекта \mathbf{A} и \mathbf{B} указанной структуры. Будем считать, что $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times n}$. Пусть $\mathbf{\Omega}^{n \times n \times n \times n}$ — это тензор, такой что его элемент Ω_{ijkl} равен квадрату разности между (i, j) -ым и (k, l) -ым элементами \mathbf{A} и \mathbf{B} :

$$\Omega_{ijkl} = (\mathbf{A}(i, j) - \mathbf{B}(k, l))^2.$$

Определение 2: Путем π между двумя объектами указанной структуры \mathbf{A} и \mathbf{B} назовем множество пар индексов элементов тензора $\mathbf{\Omega}$:

$$\pi = \{\pi\} = \{((i, j), (k, l))\}, \quad r = 1, \dots, R, \quad i, j, k, l \in \{1, \dots, n\},$$

где R — длина пути, зависящая от выбора пути. Он должен удовлетворять следующим условиям:

Частичный порядок. Для элементов пути π с фиксированными значениями i, k , если такие содержатся в этом пути, можно задать порядок: $\{((i, j_r), (k, l_r))\}_{r=1}^{\tilde{R}}$, где \tilde{R} — число таких элементов.

Граничные условия. Пусть $((i, j), (k, l)) \in \pi$, тогда для этих i, j, k, l выполняется: $((1, j), (1, l)) \in \pi$ и $((i, 1), (k, 1)) \in \pi$. Также путь π содержит в противоположные углы тензора Ω : $((1, 1), (1, 1)) \in \pi$ и $((n, n), (n, n)) \in \pi$.

Непрерывность по направлению. Для упорядоченных элементов пути π с фиксированными значениями i, k : $\{((i, j_r), (k, l_r))\}_{r=1}^{\tilde{R}}$, выполняется условие непрерывности: $((i, j_r), (k, l_r)) = (p_1, p_2)$ и $((i, j_{r-1}), (k, l_{r-1})) = (q_1, q_2)$, $r = 2, \dots, \tilde{R}$, тогда

$$p_1 - q_1 \leq 1, \quad p_2 - q_2 \leq 1.$$

Это ограничение нужно, чтобы в шаге пути π по фиксированному направлению времени участвовали только соседние элементы матрицы (включая соседние по диагонали).

Монотонность по направлению. Для упорядоченных элементов пути π с фиксированными значениями i, k : $\{((i, j_r), (k, l_r))\}_{r=1}^{\tilde{R}}$, выполняется условие непрерывности: $((i, j_r), (k, l_r)) = (p_1, p_2)$ и $((i, j_{r-1}), (k, l_{r-1})) = (q_1, q_2)$, $r = 2, \dots, \tilde{R}$. Тогда выполняется хотя бы одно из условий

$$p_1 - q_1 \geq 1, \quad p_2 - q_2 \geq 1.$$

Это ограничение обусловлено природой рассматриваемых последовательностей и предназначено для монотонности функции выравнивания времени.

Определение 3. Стоимостью $Cost(\mathbf{A}, \mathbf{B}, \pi)$ пути π между объектами \mathbf{A} и \mathbf{B} назовем

$$Cost(\mathbf{A}, \mathbf{B}, \pi) = \sum_{((i,j),(k,l)) \in \pi} \Omega_{ijkl}. \quad (2)$$

Определение 4. Путем наименьшей стоимости (выравнивающим путем) $\hat{\pi}$ между объектами \mathbf{A} и \mathbf{B} назовем путь, имеющий наименьшую стоимость среди всех возможных путей между объектами \mathbf{A} и \mathbf{B} :

$$\hat{\pi} = \underset{\pi}{\operatorname{argmin}} Cost(\mathbf{A}, \mathbf{B}, \pi). \quad (3)$$

Определение 5. Функцию расстояния $MatrixDTW(\mathbf{A}, \mathbf{B})$ между объектами \mathbf{A} и \mathbf{B} зададим как стоимость выравнивающего пути $\hat{\pi}$:

$$MatrixDTW(\mathbf{A}, \mathbf{B}) = Cost(\mathbf{A}, \mathbf{B}, \hat{\pi}). \quad (4)$$

Мотивация метода

В случае выравнивания одной временной шкалы итоговая матрица в каждой ячейке i, j содержит расстояние между под-рядом временного ряда и под-рядом временного ряда.

Рассмотрим классический алгоритм динамического выравнивания двух временных рядов $\mathbf{s} \in R^n$ и $\mathbf{c} \in R^m$:

```

 $D(1 : n + 1, 1 : m + 1) = inf;$ 
 $D(1, 1) = 0;$ 
for  $i = 2 : n + 1$ 
  for  $j = 2 : m + 1$ 
     $cost = (\mathbf{s}(i - 1) - \mathbf{c}(j - 1))^2;$ 
     $D(i, j) = cost + \min([D(i - 1, j), D(i, j - 1), D(i - 1, j - 1)]);$ 
 $DTW(\mathbf{s}, \mathbf{c}) = \sqrt{D(n + 1, m + 1)}.$ 

```

Элемент $D(i, j)$ матрицы D соответствует стоимости выравнивающего пути между подпоследовательностями исходных временных рядов: $\mathbf{s}(1 : i)$ и $\mathbf{c}(1 : j)$. Алгоритм построения наилучшего выравнивания времени подразумевает, что оптимальное выравнивание таких двух подпоследовательностей может быть получено тремя различными вариантами:

1. Минимальная из перечисленных вариантов стоимость у выравнивающего пути между $\mathbf{s}(1 : i - 1)$ и $\mathbf{c}(1 : j)$. Тогда к этому выравнивающему пути добавляется пара индексов (i, j) .
2. Минимальная из перечисленных вариантов стоимость у выравнивающего пути между $\mathbf{s}(1 : i)$ и $\mathbf{c}(1 : j - 1)$. Тогда к этому выравнивающему пути добавляется пара индексов (i, j) .
3. Минимальная из перечисленных вариантов стоимость у выравнивающего пути между $\mathbf{s}(1 : i - 1)$ и $\mathbf{c}(1 : j - 1)$. Тогда к этому выравнивающему пути добавляется пара индексов (i, j) .

Предлагается перенести эти рассуждения на случай двух временных шкал и выравнивания двух матриц \mathbf{A} и \mathbf{B} . Тогда элемент $((i, j), (k, l))$ четырехмерного тензора должен соответствовать стоимости выравнивающего пути между $\mathbf{A}(1 : i, 1 : j)$ и $\mathbf{B}(1 : k, 1 : l)$. При этом предлагается получать такое оптимальное выравнивание семью различными способами:

1. Минимальная из перечисленных вариантов стоимость у выравнивающего пути между $\mathbf{A}(1 : i - 1, 1 : j)$ и $\mathbf{B}(1 : k, 1 : l)$. Тогда к этому выравнивающему пути добавляется пара индексов $((i, j), (k, l))$, а также дублируются все пары индексов пути $\{((i - 1, j), (k, l)) \in \pi \mid j, l - \text{произвольные}, i, k - \text{фиксированные}\}$ с заменой $i - 1$ на i .
2. Минимальная из перечисленных вариантов стоимость у выравнивающего пути между $\mathbf{A}(1 : i, 1 : j - 1)$ и $\mathbf{B}(1 : k, 1 : l)$. Тогда к этому выравнивающему пути добавляется пара индексов $((i, j), (k, l))$, а также дублируются все пары индексов пути $\{((i, j - 1), (k, l)) \in \pi \mid i, k - \text{произвольные}, j, l - \text{фиксированные}\}$ с заменой $j - 1$ на j .
3. Минимальная из перечисленных вариантов стоимость у выравнивающего пути между $\mathbf{A}(1 : i, 1 : j)$ и $\mathbf{B}(1 : k - 1, 1 : l)$. Тогда к этому выравнивающему пути добавляется пара индексов $((i, j), (k, l))$, а также дублируются все пары индексов пути $\{((i, j), (k - 1, l)) \in \pi \mid j, l - \text{произвольные}, i, k - \text{фиксированные}\}$ с заменой $k - 1$ на k .
4. Минимальная из перечисленных вариантов стоимость у выравнивающего пути между $\mathbf{A}(1 : i, 1 : j)$ и $\mathbf{B}(1 : k, 1 : l - 1)$. Тогда к этому выравнивающему пути добавляется пара индексов $((i, j), (k, l))$, а также дублируются все пары индексов пути $\{((i, j), (k, l - 1)) \in \pi \mid i, k - \text{произвольные}, j, l - \text{фиксированные}\}$ с заменой $l - 1$ на l .
5. Минимальная из перечисленных вариантов стоимость у выравнивающего пути между $\mathbf{A}(1 : i - 1, 1 : j)$ и $\mathbf{B}(1 : k - 1, 1 : l)$. Тогда к этому выравнивающему пути добавляется пара индексов $((i, j), (k, l))$, а также дублируются все пары индексов пути $\{((i - 1, j), (k - 1, l)) \in \pi \mid j, l - \text{произвольные}, i, k - \text{фиксированные}\}$ с заменой $i - 1$ на i , $k - 1$ на k .
6. Минимальная из перечисленных вариантов стоимость у выравнивающего пути между $\mathbf{A}(1 : i, 1 : j - 1)$ и $\mathbf{B}(1 : k, 1 : l - 1)$. Тогда к этому выравнивающему пути добавляется пара индексов $((i, j), (k, l))$, а также дублируются все пары индексов

пути $\{((i, j-1), (k, l-1)) \in \pi \mid i, k - \text{произвольные}, j, l - \text{фиксированные}\}$ с заменой $j-1$ на j , $l-1$ на l .

7. Минимальная из перечисленных вариантов стоимость у выравнивающего пути между $\mathbf{A}(1 : i-1, 1 : j-1)$ и $\mathbf{B}(1 : k-1, 1 : l-1)$. Тогда к этому выравнивающему пути добавляется пара индексов $((i, j), (k, l))$, а также дублируются все пары индексов пути $\{((i, j-1), (k, l-1)) \in \pi \mid i, k - \text{произвольные}, j, l - \text{фиксированные}\}$ с заменой $j-1$ на j , $l-1$ на l и все пары индексов пути $\{((i-1, j), (k-1, l)) \in \pi \mid j, l - \text{произвольные}, i, k - \text{фиксированные}\}$ с заменой $i-1$ на i , $k-1$ на k .

С учетом указанных вариантов алгоритм вычисления расстояния между объектами примет вид:

```

D(1 : n + 1, 1 : n + 1, 1 : n + 1, 1 : n + 1) = inf;
D(1, 1, 1, 1) = 0;
π(1, 1, 1, 1) = ((1, 1), (1, 1))
for i = 2 : n + 1
    for j = 2 : n + 1
        for k = 2 : n + 1
            for l = 2 : n + 1
                 $\overline{i, j, k, l} = \text{argmin}([D(i-1, j, k, l), D(i, j-1, k, l),$ 
                     $D(i, j, k-1, l), D(i, j, k, l-1), D(i-1, j, k-1, l), D(i, j-1, k, l-1),$ 
                     $D(i-1, j-1, k-1, l-1)])$ ;
                if  $\overline{i, j, k, l} == i-1, j, k, l$  :
                    cost =  $\sum_{r, f: ((i-1, r), (k, f)) \in \pi(i-1, j, k, l)} (\mathbf{A}(i, r) - \mathbf{B}(k, f))^2$ ;
                     $\pi(i, j, k, l) = \pi(i-1, j, k, l) \cup \{((i, r), (k, f)) | ((i-1, r), (k, f)) \in$ 
                         $\pi(i-1, j, k, l)\} \cup \{((i, j), (k, l))\}$ 
                     $D(i, j, k, l) = \text{cost} + D(\overline{i, j, k, l})$ 
                if  $\overline{i, j, k, l} == i, j-1, k, l$  :
                    cost =  $\sum_{r, f: ((r, j-1), (f, l)) \in \pi(i, j-1, k, l)} (\mathbf{A}(r, j) - \mathbf{B}(f, l))^2$ ;
                     $\pi(i, j, k, l) = \pi(i, j-1, k, l) \cup \{((r, j), (f, l)) | ((r, j-1), (f, l)) \in$ 
                         $\pi(i, j-1, k, l)\} \cup \{((i, j), (k, l))\}$ 
                     $D(i, j, k, l) = \text{cost} + D(\overline{i, j, k, l})$ 
                if  $\overline{i, j, k, l} == i, j, k-1, l$  :
                    cost =  $\sum_{r, f: ((i, r), (k-1, f)) \in \pi(i, j, k-1, l)} (\mathbf{A}(i, r) - \mathbf{B}(k, f))^2$ ;
                     $\pi(i, j, k, l) = \pi(i, j, k-1, l) \cup \{((i, r), (k, f)) | ((i, r), (k-1, f)) \in$ 

```

$$\begin{aligned}
& \in \pi(i, j, k-1, l) \cup \{((i, j), (k, l))\} \\
D(i, j, k, l) &= cost + D(\overline{i, j, k, l}) \\
if \quad \overline{i, j, k, l} &= i, j, k, l-1 : \\
cost &= \sum_{r, f: ((r, j), (f, l-1)) \in \pi(i, j, k, l-1)} (\mathbf{A}(r, j) - \mathbf{B}(f, l))^2; \\
\pi(i, j, k, l) &= \pi(i, j, k, l-1) \cup \{((r, j), (f, l)) | ((r, j), (f, l-1)) \in \\
&\in \pi(i, j, k, l-1) \cup \{((i, j), (k, l))\} \\
D(i, j, k, l) &= cost + D(\overline{i, j, k, l}) \\
if \quad \overline{i, j, k, l} &= i-1, j, k-1, l : \\
cost &= \sum_{r, f: ((i-1, r), (k-1, f)) \in \pi(i-1, j, k-1, l)} (\mathbf{A}(i, r) - \mathbf{B}(k, f))^2; \\
\pi(i, j, k, l) &= \pi(i-1, j, k-1, l) \cup \{((i, r), (k, f)) | \\
&|((i-1, r), (k-1, f)) \in \pi(i-1, j, k-1, l) \cup \{((i, j), (k, l))\} \\
D(i, j, k, l) &= cost + D(\overline{i, j, k, l}) \\
if \quad \overline{i, j, k, l} &= i, j-1, k, l-1 : \\
cost &= \sum_{r, f: ((r, j-1), (f, l-1)) \in \pi(i, j-1, k, l-1)} (\mathbf{A}(r, j) - \mathbf{B}(f, l))^2; \\
\pi(i, j, k, l) &= \pi(i, j-1, k, l-1) \cup \{((r, j), (f, l)) | \\
&|((r, j-1), (f, l-1)) \in \pi(i, j-1, k, l-1) \cup \{((i, j), (k, l))\} \\
D(i, j, k, l) &= cost + D(\overline{i, j, k, l}) \\
if \quad \overline{i, j, k, l} &= i-1, j-1, k-1, l-1 : \\
cost &= \sum_{r, f: ((r, j-1), (f, l-1)) \in \pi(i, j-1, k, l-1)} (\mathbf{A}(r, j) - \mathbf{B}(f, l))^2 + \\
&+ \sum_{r, f: ((i-1, r), (k-1, f)) \in \pi(i-1, j, k-1, l)} (\mathbf{A}(i, r) - \mathbf{B}(k, f))^2; \\
\pi(i, j, k, l) &= \pi(i, j-1, k, l-1) \cup \{((r, j), (f, l)) | \\
&|((r, j-1), (f, l-1)) \in \pi(i, j-1, k, l-1) \cup \{((i, r), (k, f))\} \\
&|((i-1, r), (k-1, f)) \in \pi(i-1, j, k-1, l) \cup \{((i, j), (k, l))\} \\
D(i, j, k, l) &= cost + D(\overline{i, j, k, l})
\end{aligned}$$

$$MatrixDTW(\mathbf{A}, \mathbf{B}) = \sqrt{D(n+1, n+1, n+1, n+1)}$$

Следует отметить, что алгоритм имеет высокую сложность вычисления — $O(n^4)$. Предполагается ускорение метода при помощи использования аналогичного Sakoe-Chiba band ограничения для случая двух осей времени. Такой подход позволит сократить вычислительную сложность алгоритма до $O(n^2k^2)$, где k — заранее заданное число.

4 Вычислительный эксперимент

Вычислительный эксперимент проводился как на модельных данных, порождение которых согласуется с гипотезой порождения данных, так и на реальных данных: кол-

лекция MNIST, к которым применялись различного рода искажения, согласующиеся с гипотезой порождения данных.

В качестве модельных данных использовались матрицы, в которых ненулевыми были: горизонтальная и вертикальная линии, случайный прямоугольник. На модельные данные накладывался шум. К данным применялось случайное количество преобразований времени: как строки, так и столбцы матрицы могли быть продублированы, тем самым создавая новый объект. Такие искажения согласуются с гипотезой о наличии локальных искажений. Также к матрицам могли быть конкатенированы разное число нулевых строк и столбцов, что согласуется с гипотезой о наличии глобальных искажений в данных.

Как было указано в постановке задачи, решается задача метрической классификации методами ближайшего соседа.

Данные / Метод	L_2	MatrixDTW
Модельные данные без искажений	92%	100%
Модельные данные с искажениями	86%	100%
Модельные данные с искажениями и шумом	69%	92%

Была создана подвыборка выборки MNIST, содержащая по 100 объектов двух классов: 0 и 1. К ней были применены описанные выше искажения, что увеличило в 3 раза объем выборки.

Данные / Метод	L_2	MatrixDTW
MNIST без искажений	95%	95%
MNIST с искажениями	53%	92%

Ниже на рисунке приведен пример оптимального выравнивания для различных объектов:

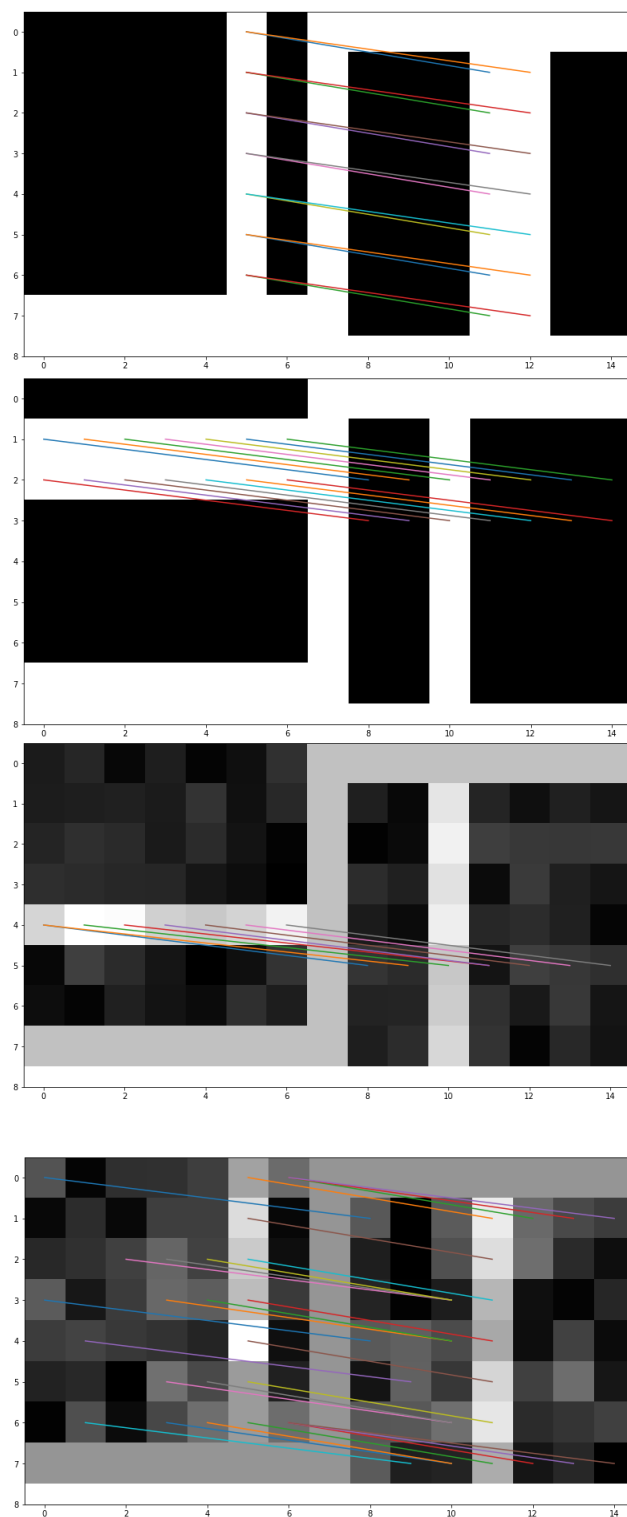


Рис. 1: Визуализация выравнивающего пути для модельных объектов двух классов.

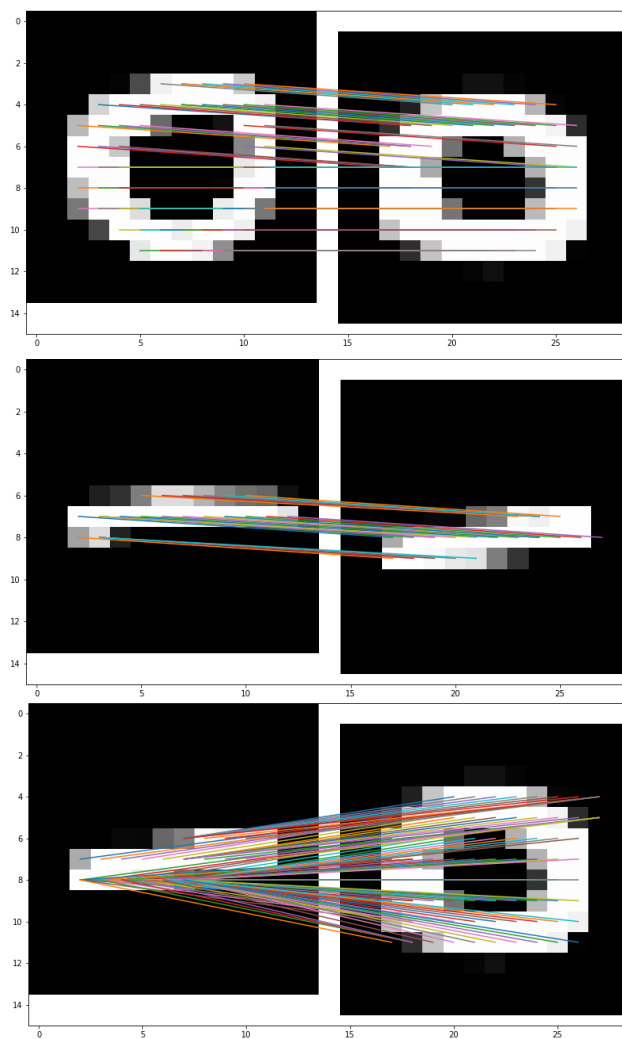


Рис. 2: Визуализация выравнивающего пути для объектов классов 0 и 1.