Функции расстояния



Измеряй измеримое

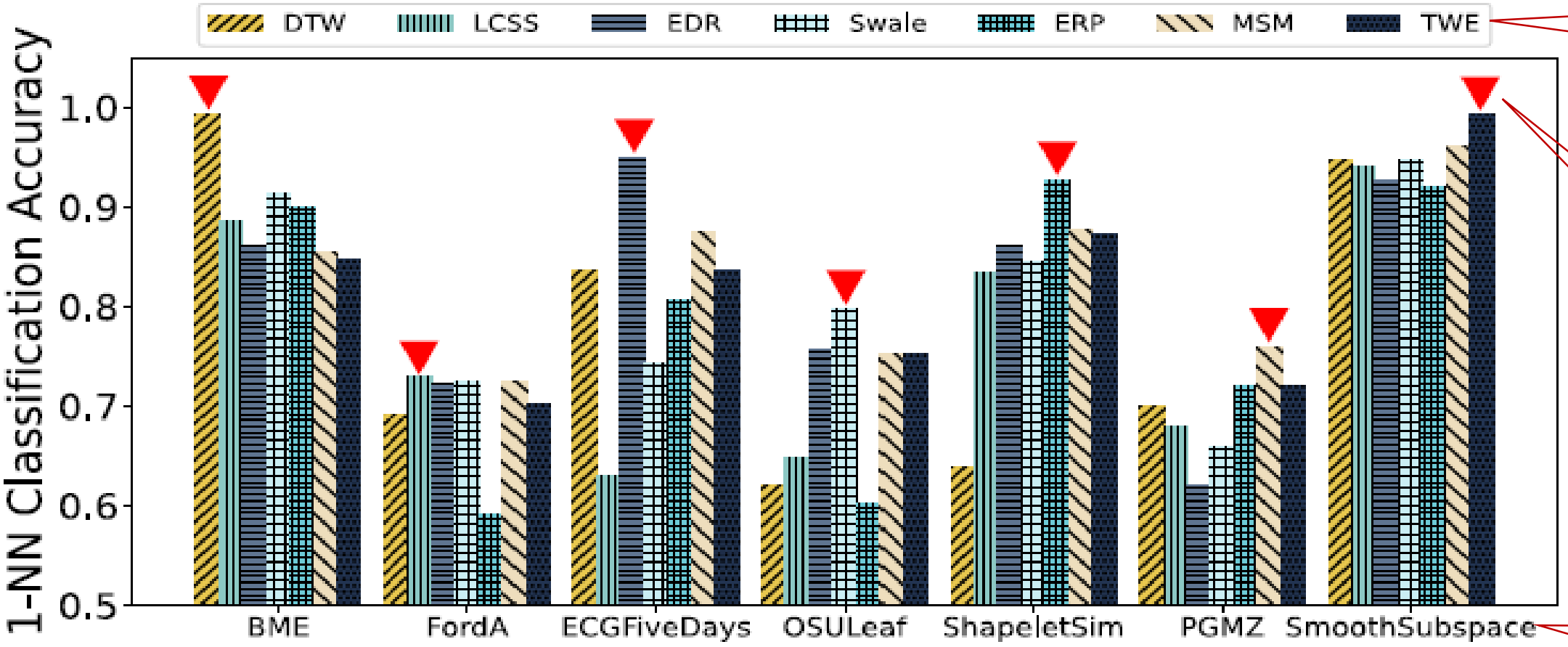
и делай неизмеримое измеримым.

Г. Галилей

Содержание

- Иерархия функций расстояния
- Евклидово расстояние
- Динамическая трансформация времени (DTW)

Важность выбора функции расстояния



Функции расстояния

Лучшие результаты

Наборы данных

Paparrizos, J. *et al.* Accelerating similarity search for elastic measures: A study and new generalization of lower bounding distances. Proc. VLDB Endow. 16(8), 2019–2032 (2023). DOI 10.14778/3594512.3594530. https://www.vldb.org/pvldb/vol16/p2019-paparrizos.pdf

Аксиомы метрики: Dist: $M \times M \to \mathbb{R}$: $\forall x, y, z \in M$

- Аксиома тождества:
$$\mathrm{Dist}(x, x) = 0$$
- Аксиома положительности:
$$\mathrm{Dist}(x, y) \geq 0$$
- Аксиома симметричности:
$$\mathrm{Dist}(x, y) = \mathrm{Dist}(y, x)$$
- Аксиома (неравенство) треугольника:
$$\mathrm{Dist}(x, z) \leq \mathrm{Dist}(x, y) + \mathrm{Dist}(y, z)$$

Метрика

Функция расстояния

Ослабленная функция расстояния

Иерархия функций расстояния



Содержание

- Иерархия функций расстояния
- **Евклидово расстояние**
  - Определение
  - Важность неравенства треугольника
  - Z-нормализованное евклидово расстояние
  - Преимущества и недостатки
- Динамическая трансформация времени (DTW)

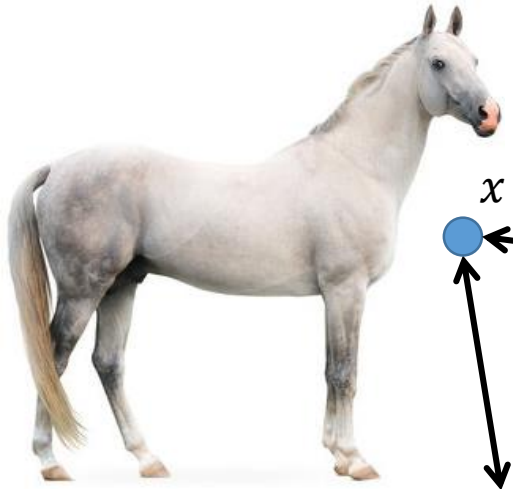
Евклидово расстояние*: $\mathrm{ED}(Q,C)=\sqrt{\sum_{i=1}^{m}(q_i-c_i)^2}$

- Интуитивно понятное расстояние
  - геометрический смысл – удаленность объектов друг от друга
  - сравнение равных по длине рядов по принципу «один-к-одному» без учета их формы
- Вычислительная сложность $O(m)$
- Ускорение вычислений: $\mathrm{ED}^2(Q,C)=\sum_{i=1}^{m}(q_i-c_i)^2$
  - Вычисляется быстрее, но не метрика (не выполняется неравенство треугольника)
  - Усиление несхожести
  - Усложнение интерпретации (квадраты единиц)



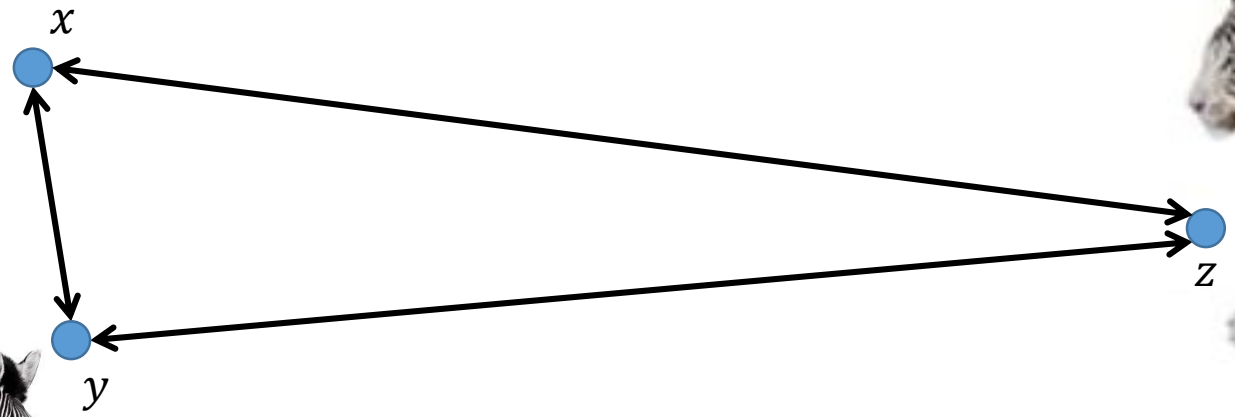
Евклид*
325-265 д.н.э.

Неравенство треугольника выполнено



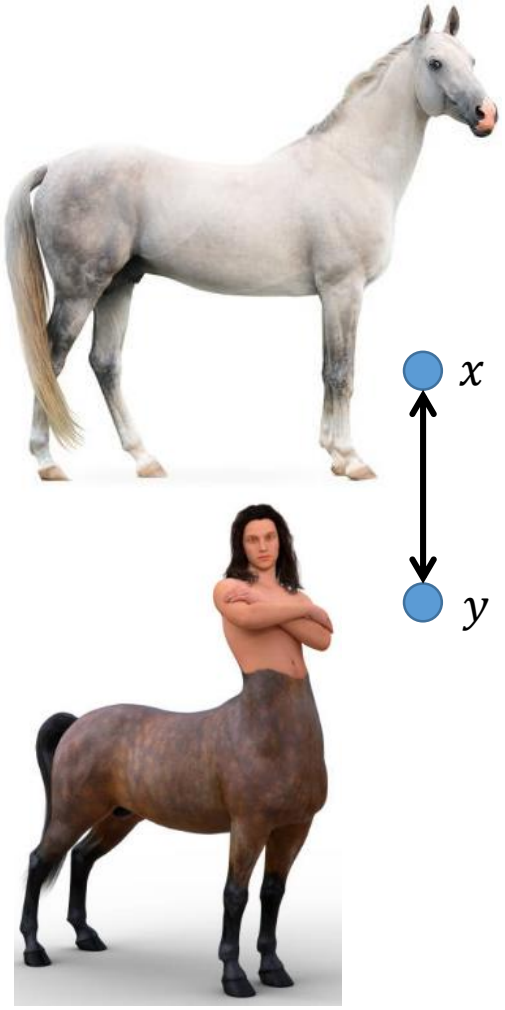


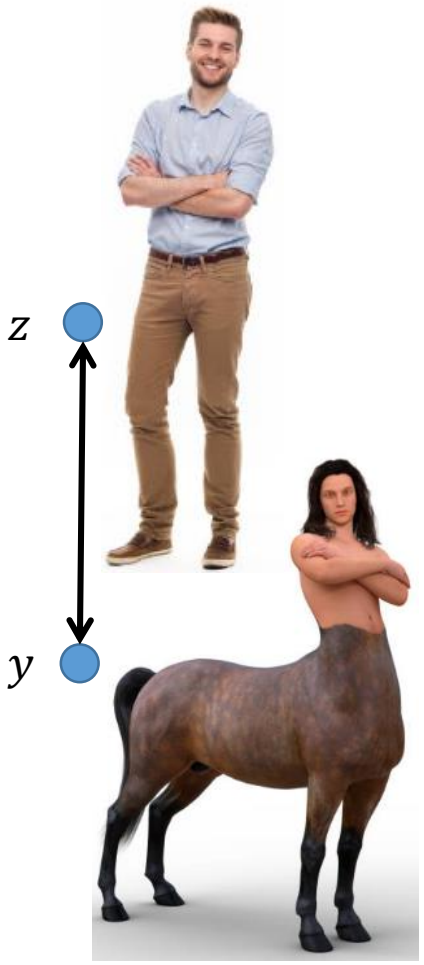




$$\forall x, y, z$$
$$\mathrm{Dist}(x, z) \leq \mathrm{Dist}(x, y) + \mathrm{Dist}(y, z)$$

Неравенство треугольника нарушено: ОНО РАБОТАЕТ, НО НЕ ВСЕГДА





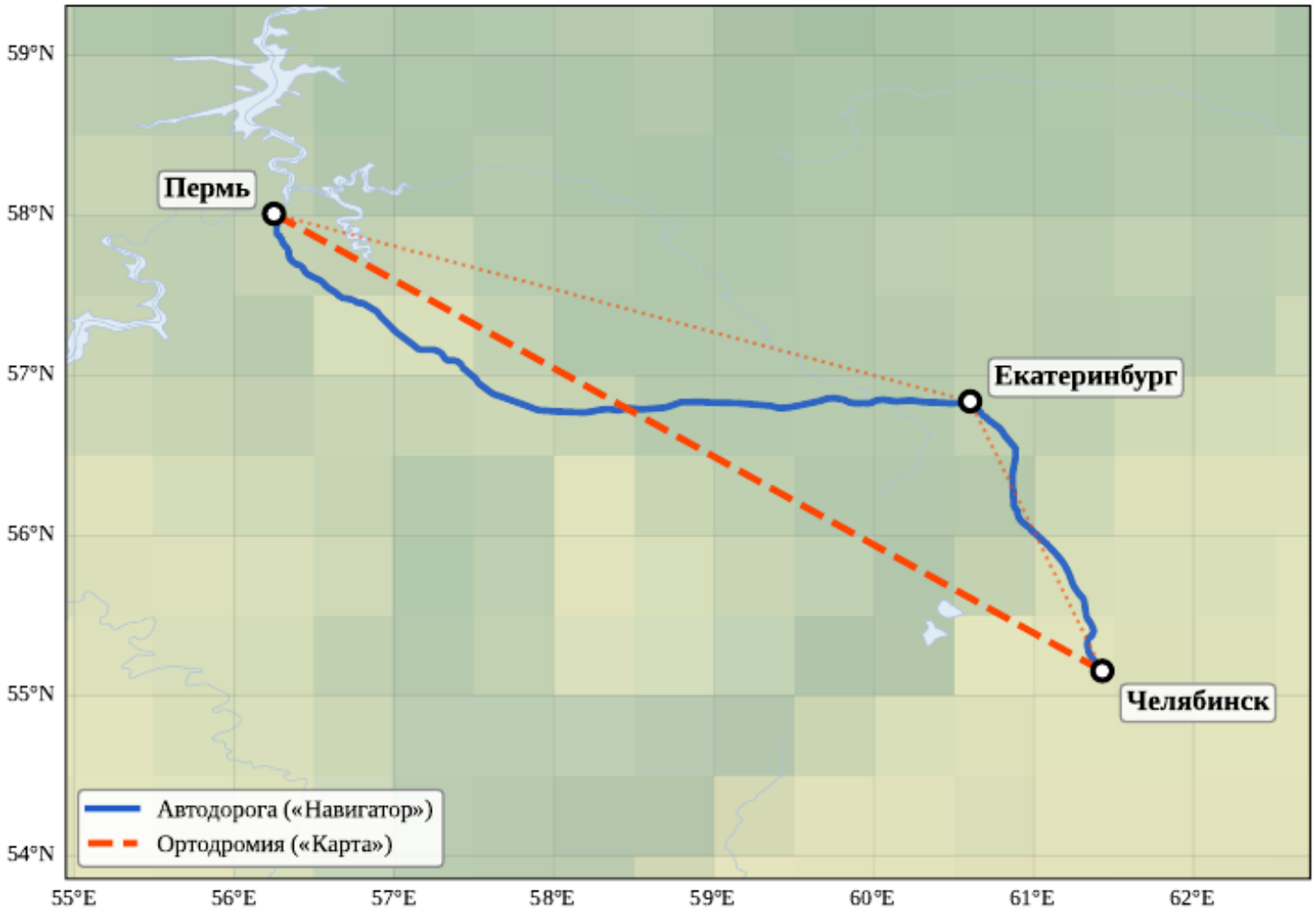


$$\neg(\forall x, y, z \; \mathrm{Dist}(x, z) \leq \mathrm{Dist}(x, y) + \mathrm{Dist}(y, z))$$

$$\Downarrow$$

$$\exists\, x, y, z \; \mathrm{Dist}(x, z) > \mathrm{Dist}(x, y) + \mathrm{Dist}(y, z)$$

Нарушение неравенства треугольника – не повод отвергнуть функцию расстояния



$$\text{Карта}(\text{Чел}, \text{Прм}) \leq \text{Карта}(\text{Чел}, \text{Екб}) + \text{Карта}(\text{Екб}, \text{Прм})$$

$$\text{Навигатор}(\text{Чел}, \text{Прм}) > \text{Навигатор}(\text{Чел}, \text{Екб}) + \text{Навигатор}(\text{Екб}, \text{Прм})$$

Выполнение неравенства треугольника – возможность ускорить поиск ближайших соседей

- Неравенство треугольника:

$$\forall x, y, z\ \mathrm{Dist}(x, z) \leq \mathrm{Dist}(x, y) + \mathrm{Dist}(y, z)$$

- Обратное неравенство треугольника

$$\forall x, y, z\ |\mathrm{Dist}(x, y) - \mathrm{Dist}(x, z)| \leq \mathrm{Dist}(y, z)$$

- Нижняя граница расстояния

Если x фиксирован, y, z – произвольные, то LB – нижняя граница $\mathrm{Dist}(y, z)$

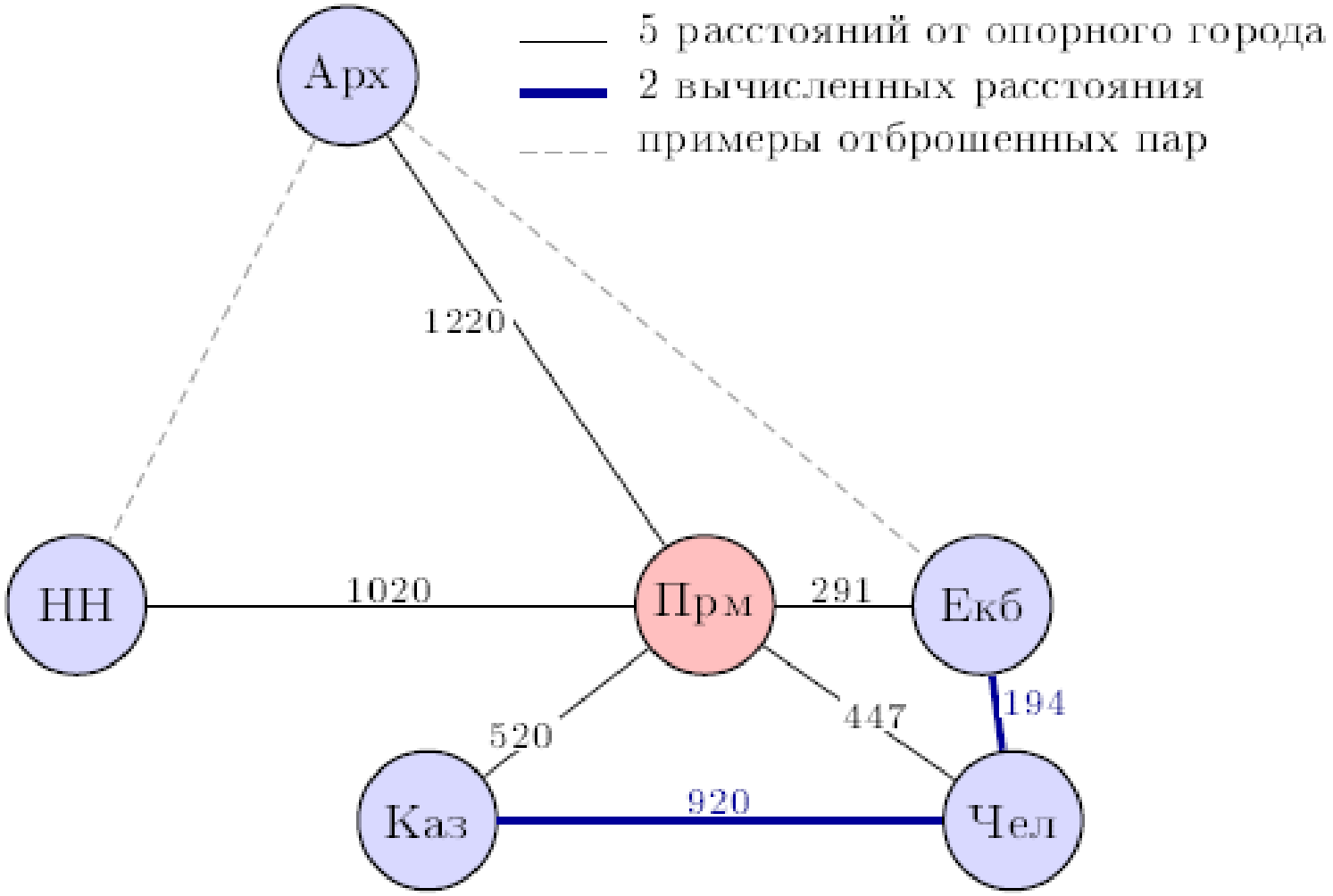
$$\mathrm{LB}(y, z) = |\mathrm{Dist}(x, y) - \mathrm{Dist}(x, z)|$$

- Контрапозиция обратного неравенства треугольника
  - bsf – текущий минимум расстояния между перебираемыми парами
  - Если $\mathrm{LB}(y, z) > bsf$, то $\mathrm{Dist}(y, z) > bsf$. Тогда пару (y, z) можно отбросить без вычисления $\mathrm{Dist}(y, z)$

Неравенство треугольника ускоряет поиск ближайших соседей

Карта(·, ·)	Арх	Екб	Каз	НН	Прм	Чел
Арх	0	1690	1360	1160	1220	1960
Екб	1690	0	800	1260	291	194
Каз	1360	800	0	390	520	920
НН	1160	1260	390	0	1020	1400
Прм	1220	291	520	1020	0	447
Чел	1960	194	920	1400	447	0

Алгоритм	Количество операций
Неравенство треугольника, просмотр по возрастанию	**7**
Неравенство треугольника, просмотр по убыванию	8
Неравенство треугольника, просмотр в случайном порядке	9
Полный перебор	15



Z-нормализация: сравнение рядов без учета разницы амплитуд

$$\hat{T} = (\hat{t}_1, \dots, \hat{t}_m), \qquad \hat{t}_i = \frac{t_i - \mu}{\sigma}$$

$$\mu = \frac{1}{m}\sum_{i=1}^{m} t_i, \qquad \sigma = \sqrt{\frac{1}{m}\sum_{i=1}^{m} t_i^2 - \mu^2}$$

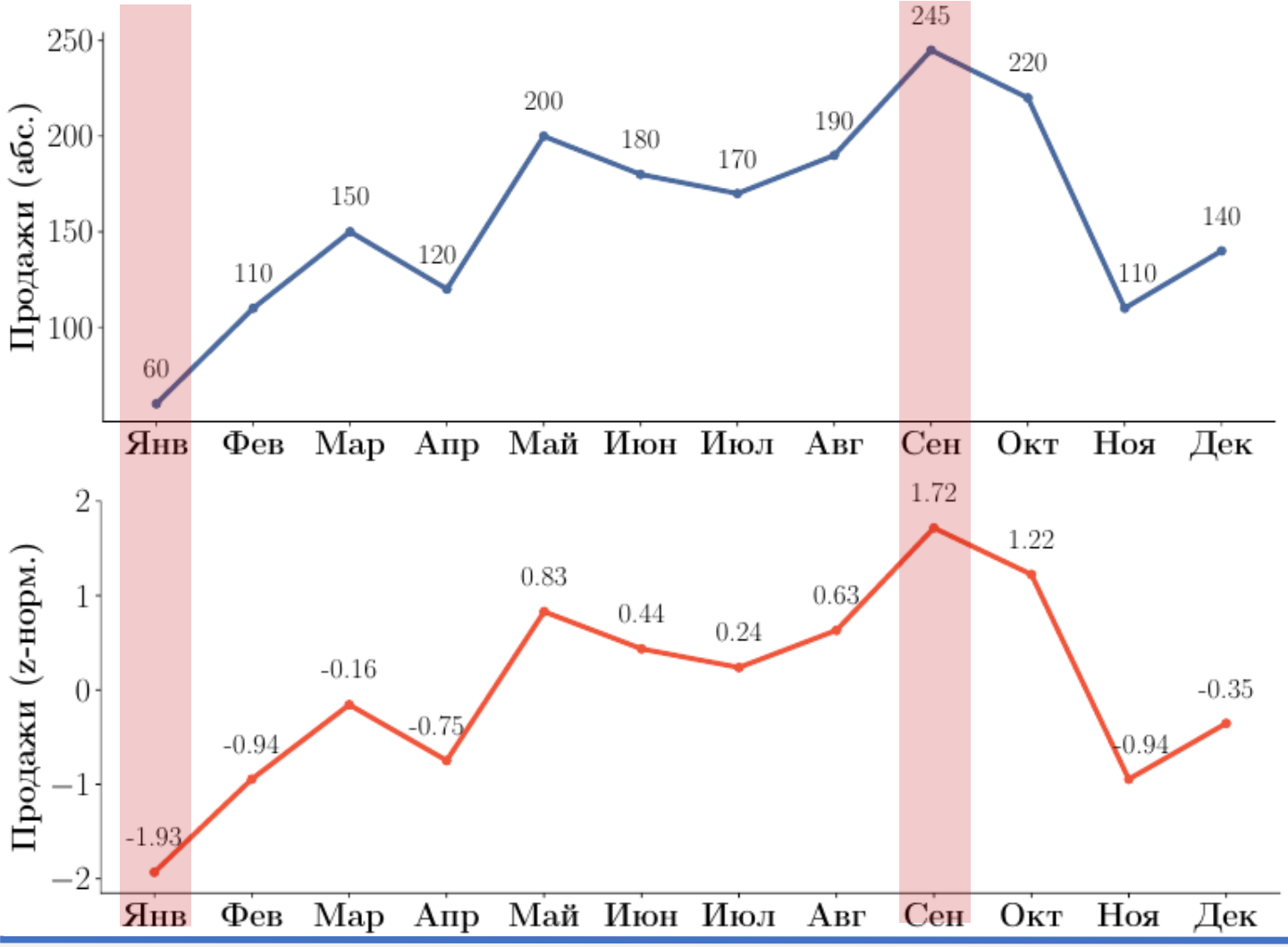
$\mu_{\hat{T}}=0$

$\sigma_{\hat{T}}=1$

$$\widehat{\mathrm{ED}}(Q, C) = \mathrm{ED}(\hat{Q}, \hat{C})$$

- Возможность сравнения рядов (подпоследовательностей) с разными масштабами и единицами измерения
- Простота интерпретации
- Улучшение обучения ML-моделей

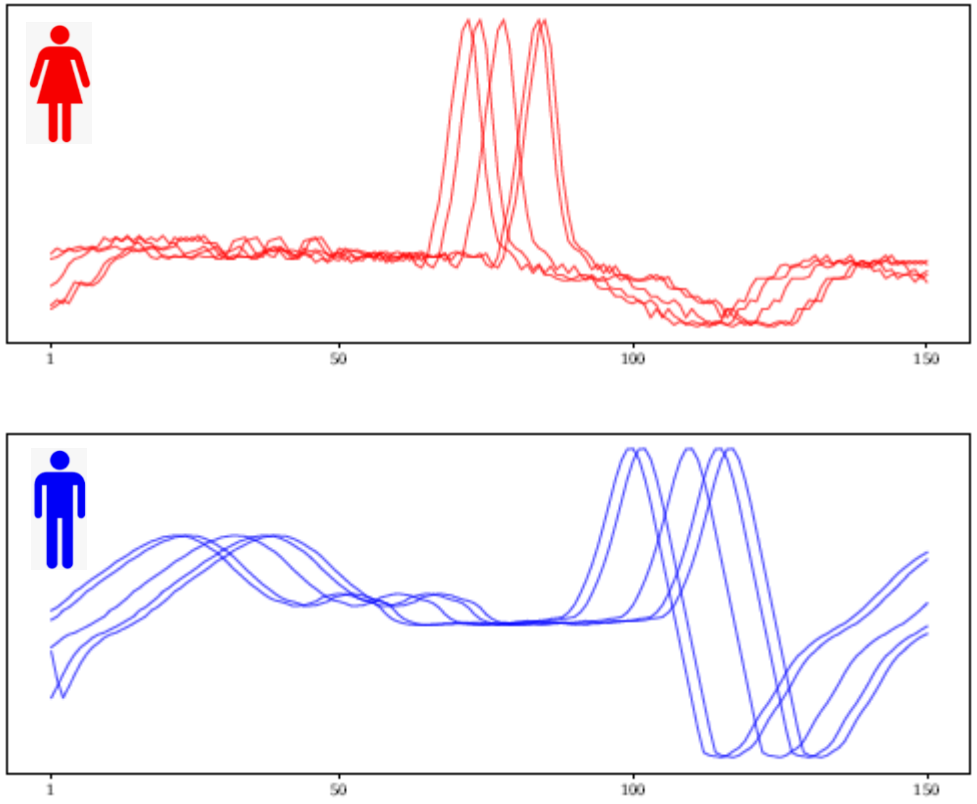
Пример z-нормализации



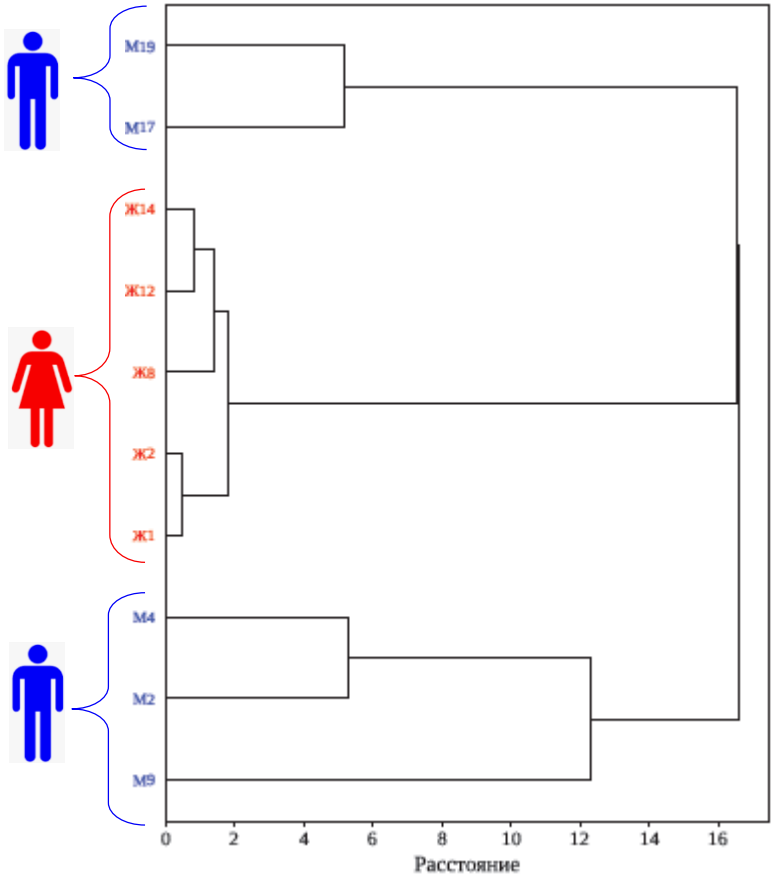
- Шкала меняется, но форма остается
- В январе продажи на 1.93 стандартных отклонения ниже среднего (значительно ниже обычного)
- В сентябре продажи на 1.72 стандартных отклонения выше среднего (значительно выше обычного)

Важность z-нормализации подпоследовательностей

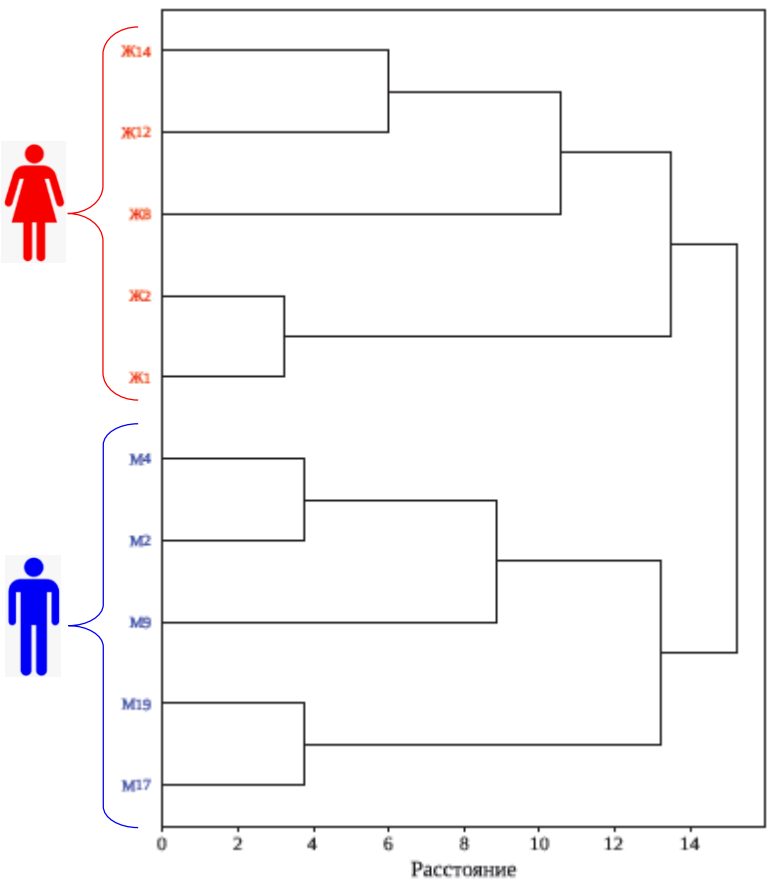
BIDMC Congestive Heart Failure Database*



Не нормализованные



Z-нормализованные

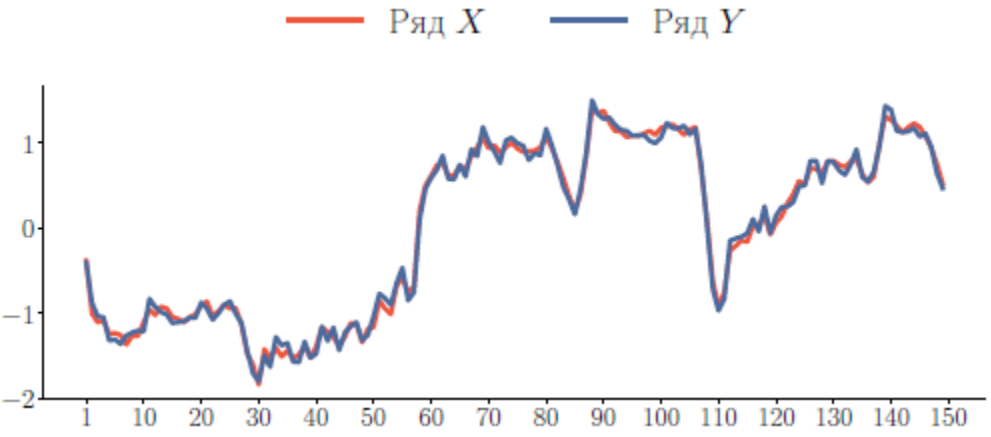


* BIDMC Congestive Heart Failure Database. URL: https://www.kaggle.com/datasets/shymammoth/bidmc-congestive-heart-failure

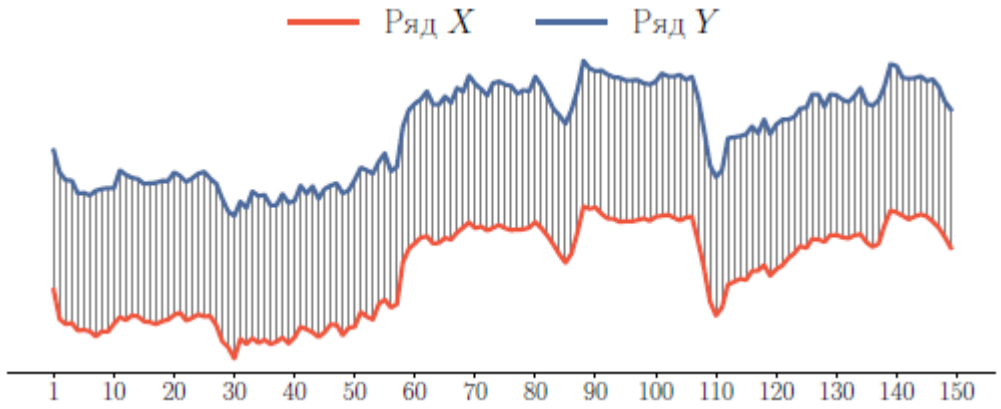
Евклидово расстояние: обсуждение

- Преимущества
  - Одна из наиболее распространенных метрик
  - Естественность (геометрическая интерпретация)
  - Выполнение неравенства треугольника (нижние границы)
- Недостатки
  - Чувствительность к выбросам
  - Чувствительность к масштабу
  - Только для рядов равной длины
  - Не учитывает сдвиги по времени

Евклидово расстояние не учитывает сдвиги по времени



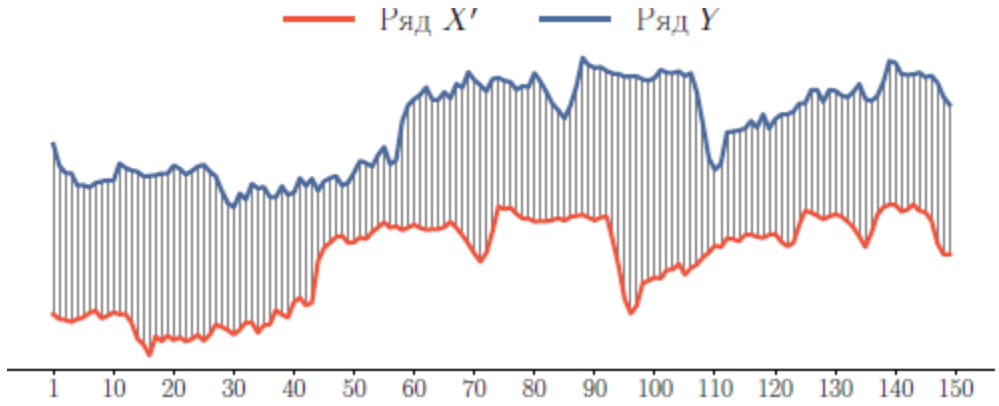
$$\mathrm{ED}(X, Y) = \mathbf{0.925}$$



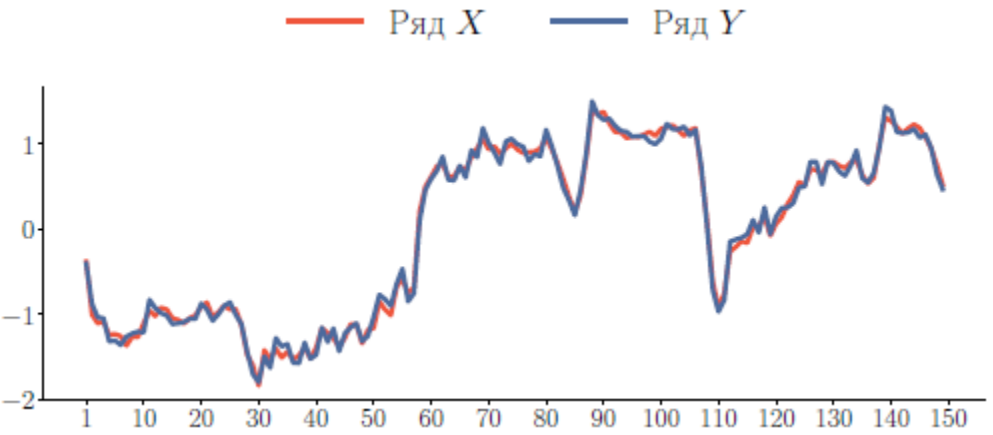
Ряд X' получен сдвигом ряда X



$$\mathrm{ED}(X', Y) = \mathbf{9.513}$$

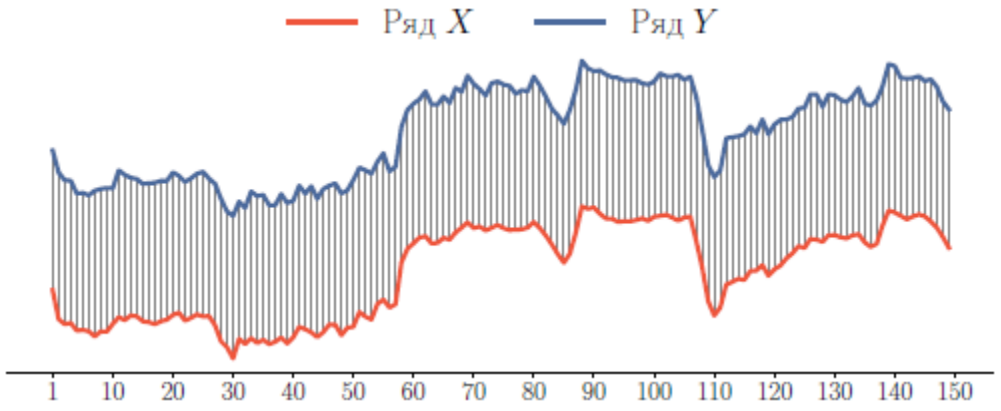


DTW-расстояние учитывает сдвиги по времени



$\mathrm{DTW}(X, Y) = \mathbf{0.838}$

$\mathrm{ED}(X, Y) = \mathbf{0.925}$

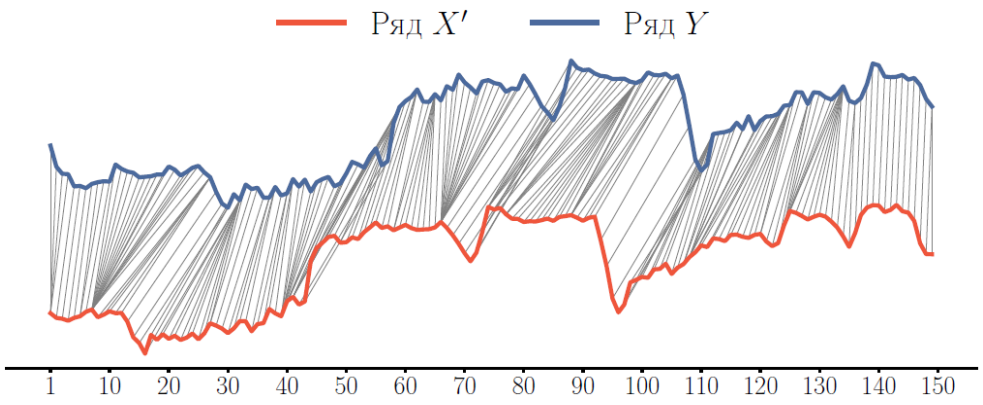


Ряд X' получен сдвигом ряда X



$\mathrm{DTW}(X', Y) = \mathbf{1.941}$

$\mathrm{ED}(X', Y) = \mathbf{9.513}$



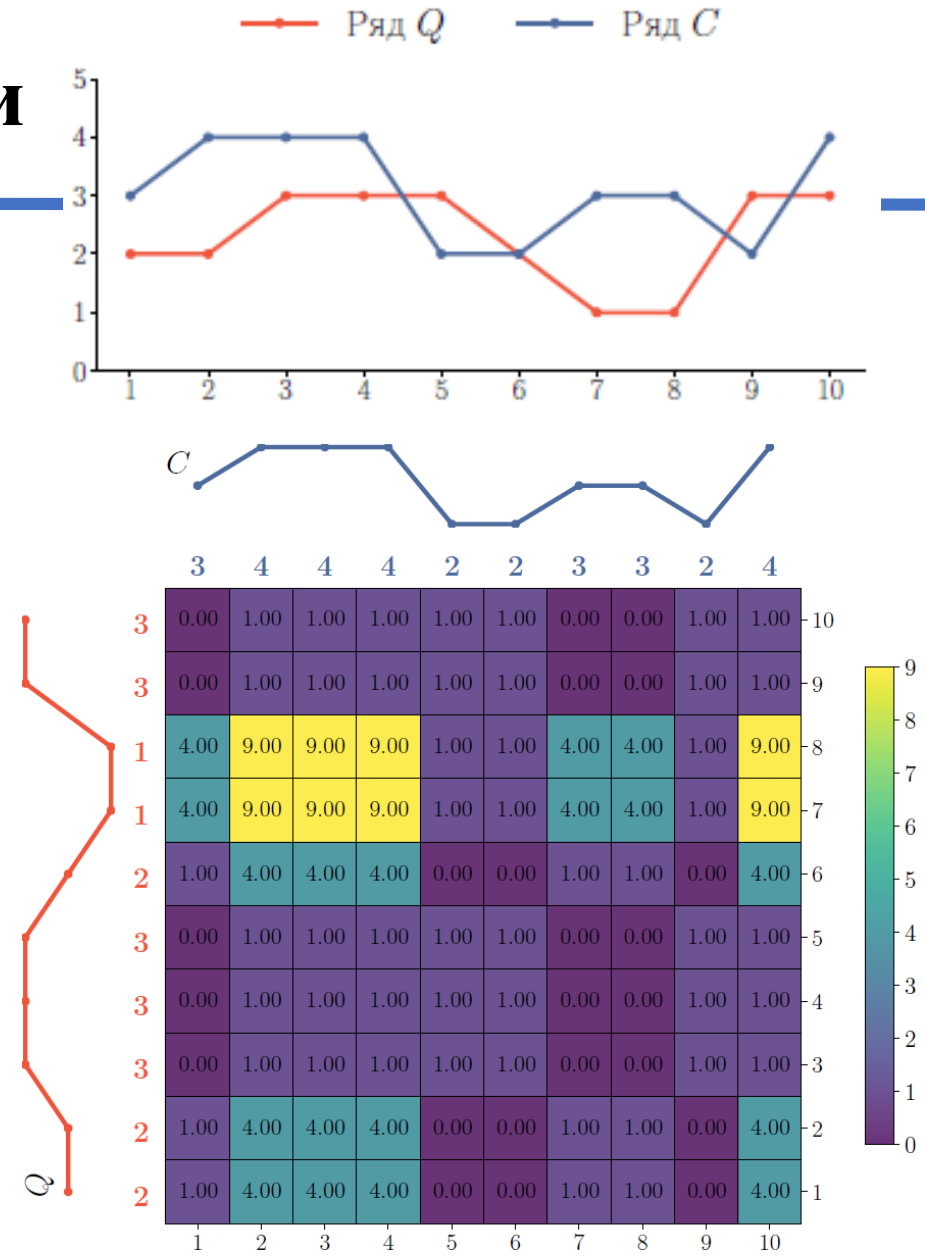
Содержание

- Иерархия функций расстояния
- Евклидово расстояние
- **Динамическая трансформация времени (DTW)**
  - Формальное определение
  - Ограничения пути трансформации

Динамическая трансформация времени, DTW (Dynamic Time Warping)

- Как DTW измеряет сходство
  - манера «один-ко-много»
  - трансформируется ось времени одного или обоих рядов так, чтобы достигнуть их оптимального взаимного выравнивания
  - вычисляется не расстояние между рядами в геометрическом смысле, а стоимость их оптимального взаимного выравнивания
- DTW – не формула, а алгоритм со многими гиперпараметрами
  - функция локальной стоимости,
  - функция стоимости пути трансформации,
  - параметры ограничений пути трансформации,
  - шаблон шага пути трансформации
  - …

Функция и матрица локальной стоимости



- *Функция локальной стоимости* измеряет стоимость выравнивания (расстояние) между точками ряда – гиперпараметр DTW

$$\mathrm{Cost}(q_i, c_j) = \mathrm{ED}^2(q_i, c_j) = (q_i - c_j)^2$$

- *Матрица локальной стоимости* (матрица расстояний) показывает стоимость выравнивания между всеми возможными парами точек рядов Q и C

$$d(i,j) = \mathrm{Cost}(q_i, c_j)$$

Путь трансформации

- Последовательность шагов по ячейкам матрицы локальной стоимости, шаг – пара (строка, столбец)

  $W = \{w_k\}_{k=1}^{K}, w_k = (i, j), m \le K < 2m - 1$

- Ограничения пути трансформации
  1. **Граничные условия** (содержит все точки рядов)

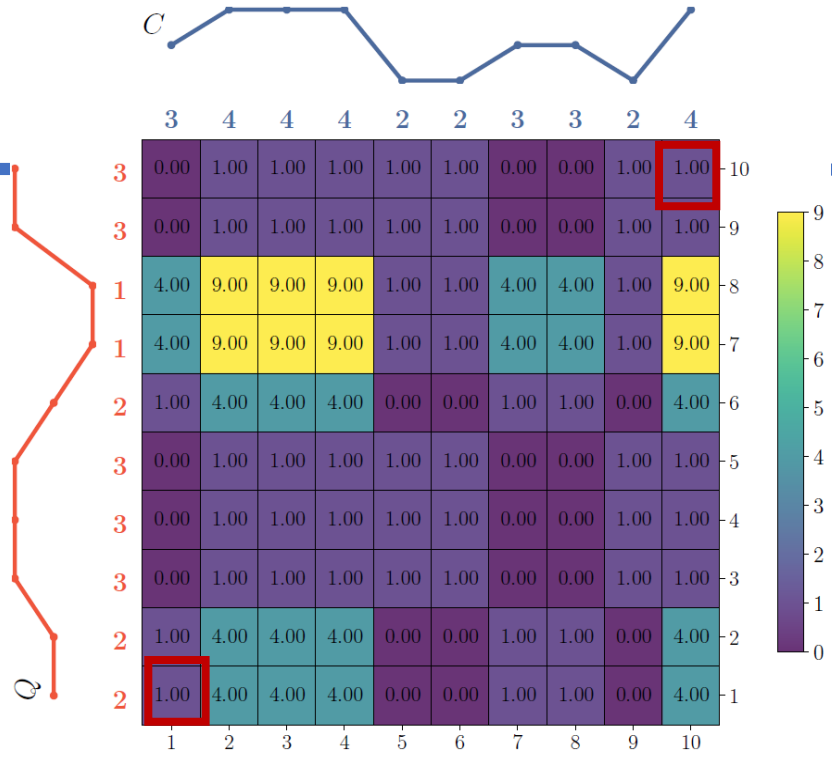
     $w_1 = (1,1), w_K = (m, m)$

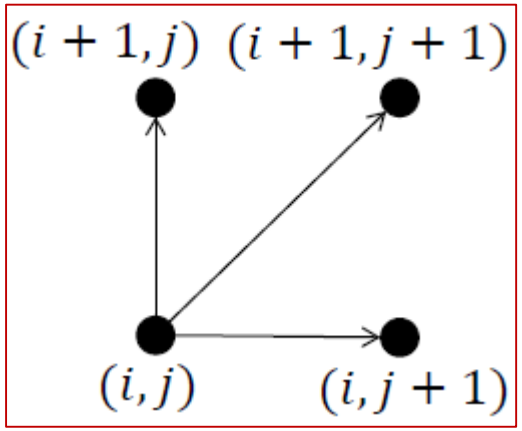
  2. **Монотонность** (без возврата к пройденному)

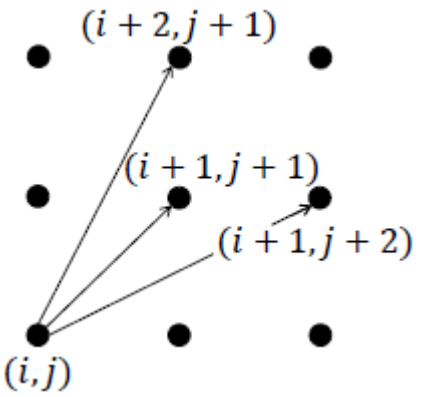
     $\forall\, w_k = (i, j)$ и $w_{k+1} = (p, q)$:

     $i - p \ge 0$ и $q - j \ge 0$

  3. **Непрерывность** (каждый шаг по шаблону)







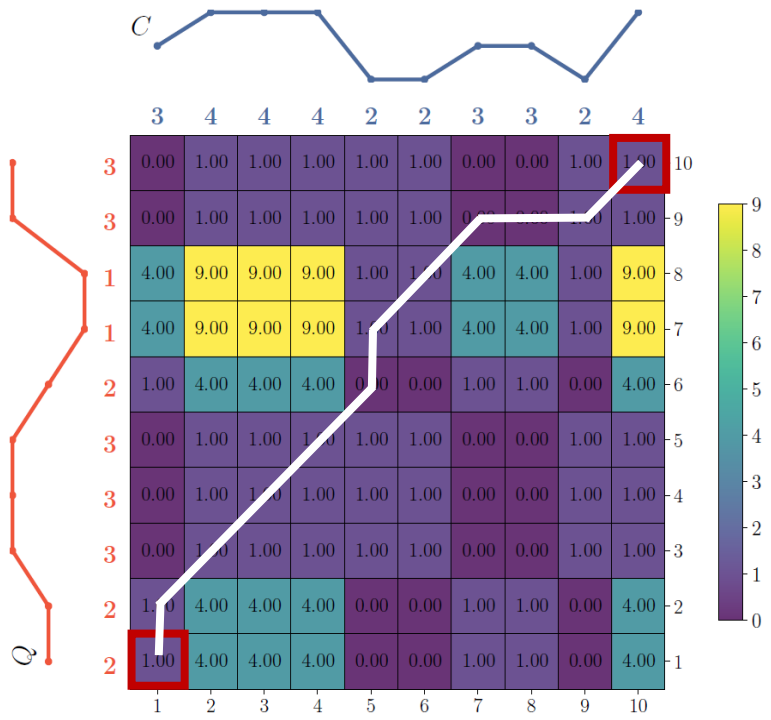
Функция стоимости пути трансформации

- Стоимость трудоемкости взаимного выравнивания рядов, квадратный корень из суммы локальных стоимостей выравнивания всех точек, входящих в путь

$$\mathrm{Cost}_W(Q,C) = \sqrt{\sum_{k=1}^{K} d(w_k)}$$

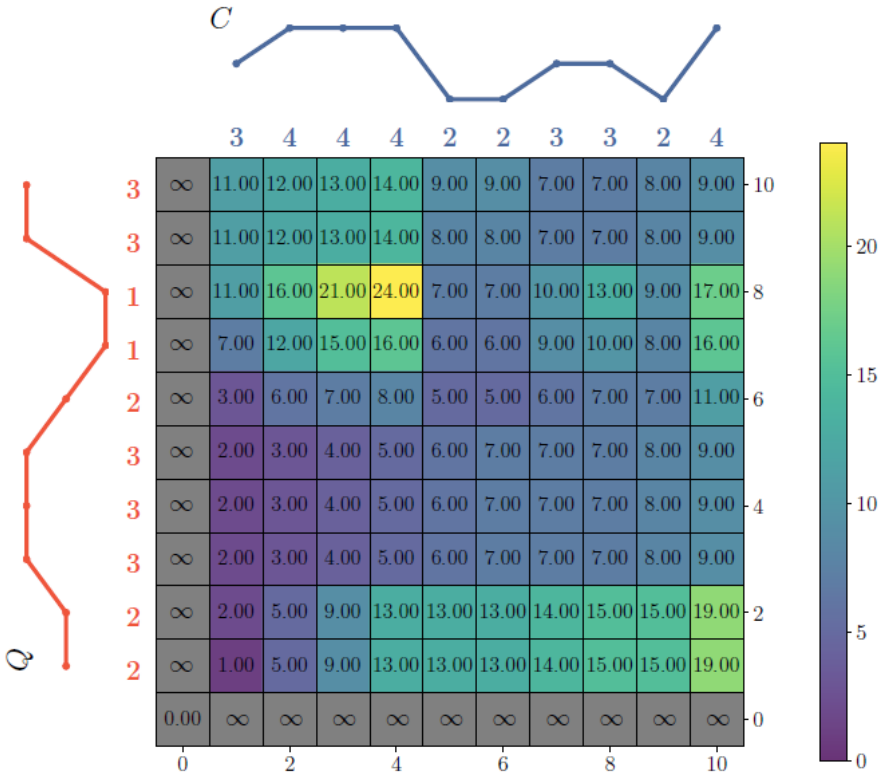
- DTW – стоимость оптимального пути трансформации W^*, имеющего минимальную стоимость

$$\mathrm{DTW}(Q,C) = \mathrm{Cost}_{W^*}(Q,C) = \min_W \sqrt{\sum_{k=1}^{K} d(w_k)}$$

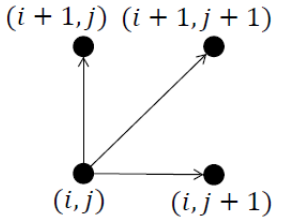


Матрица глобальной стоимости (матрица трансформации)

- Поиск оптимального пути в матрице локальной стоимости невозможен из-за экспоненциального роста числа путей-кандидатов
- Поиск оптимального пути выполняется в *матрице глобальной стоимости* методом динамического программирования



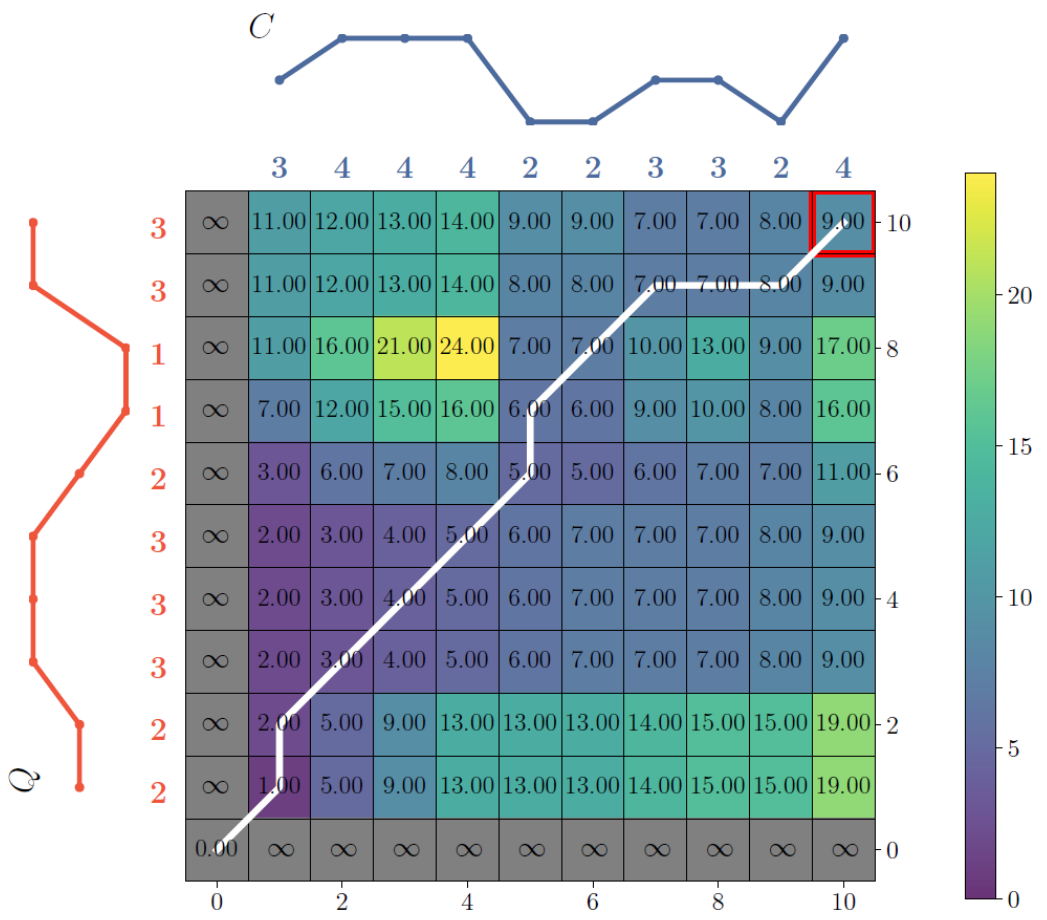
$$D(i,0) = D(0,j) = +\infty, D(0,0) = 0,$$
$$D(i,j) = d(i,j) + \min(D(i-1,j-1), D(i-1,j), D(i,j-1))$$



DTW

- Расстояние DTW между заданными рядами – это квадратный корень из значения в ячейке матрицы глобальной стоимости в финальной точке оптимального пути трансформации

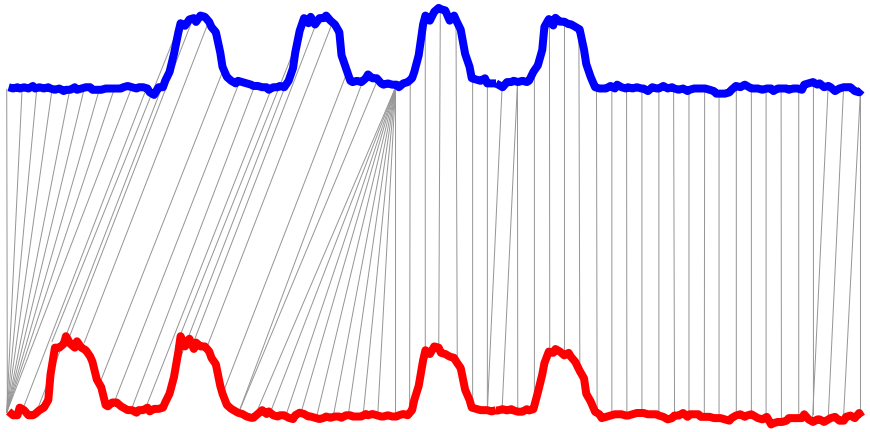
$$\mathrm{DTW}(Q, C) = \sqrt{D(m, m)}$$



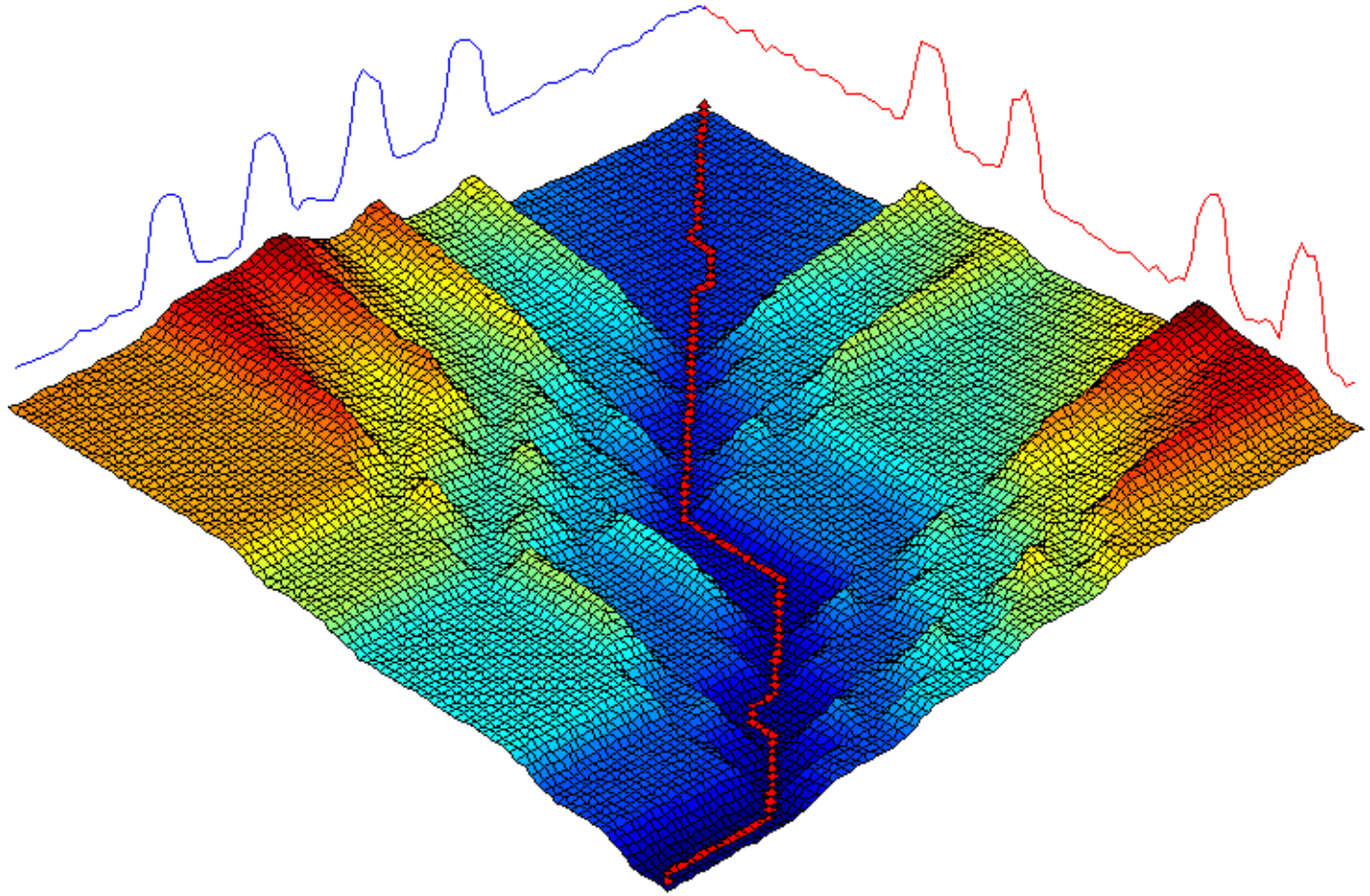
Пример оптимального пути трансформации

Недельное энергопотребление вычислительного центра (Голландия, 1997)*

C: 4-дневная рабочая неделя,
Понедельник – выходной



Q: 4-дневная рабочая неделя,
Среда – выходной



* van Wijk J.J., van Selow R.R. Cluster and calendar based visualization of time series data. INFOVIS 1999: 4-9. DOI: 10.1109/INFVIS.1999.801851

Алгоритм вычисления DTW

Вход: ряды Q, C длины m

Выход: расстояние DTW от Q до C

Данные: матрица локальной стоимости $d \in \mathbb{R}^{m \times m}$,
матрица глобальной стоимости $D \in \mathbb{R}^{(1+m)\times(1+m)}$

1: $D(0, \cdot) \leftarrow +\infty_{1+m}$; $D(\cdot, 0) \leftarrow +\infty_{1+m}$; $D(0, 0) \leftarrow 0$
2: **for** $i \leftarrow 1$ **to** m **do**
3: **for** $j \leftarrow 1$ **to** m **do**
4: $d(i, j) \leftarrow \mathrm{Cost}(q_i, c_j)$
5: $D(i, j) \leftarrow d(i, j) + \min\big(D(i-1, j-1),\ D(i-1, j),\ D(i, j-1)\big)$
6: **end for**
7: **end for**
8: **return** $\sqrt{D(m, m)}$

Алгоритм вычисления DTW

Вход: ряды Q, C длины m

Выход: расстояние DTW от Q до C

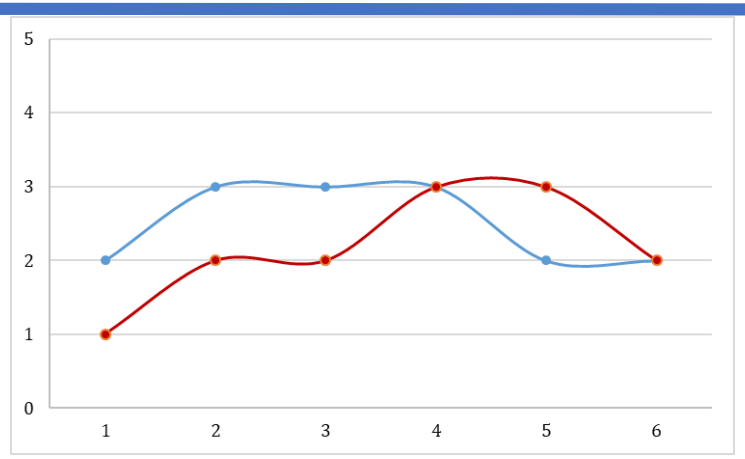
Данные: матрица локальной стоимости $d \in \mathbb{R}^{m \times m}$,
матрица глобальной стоимости $D \in \mathbb{R}^{(1+m)\times(1+m)}$

Пространственная сложность $\boldsymbol{O(m^2)}$

1: $D(0, \cdot) \leftarrow +\infty_{1+m}$; $D(\cdot, 0) \leftarrow +\infty_{1+m}$; $D(0, 0) \leftarrow 0$
2: **for** $i \leftarrow 1$ **to** m **do**
3: **for** $j \leftarrow 1$ **to** m **do**
4: $d(i, j) \leftarrow \text{Cost}(q_i, c_j)$
5: $D(i, j) \leftarrow d(i, j) + \min\big(D(i-1, j-1), D(i-1, j), D(i, j-1)\big)$
6: **end for**
7: **end for**
8: **return** $\sqrt{D(m, m)}$

Вычислительная сложность $\boldsymbol{O(m^2)}$

Пример вычисления DTW



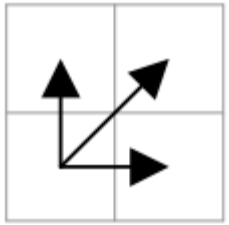
$$DTW(Q,C) = D(m,m)$$

$$D(i,j) = (q_i - c_j)^2 + \min\begin{cases} D(i-1,j) \\ D(i,j-1) \\ D(i-1,j-1) \end{cases}$$

$$D(0,0) = 0, \qquad D(i,0) = D(0,j) = +\infty$$

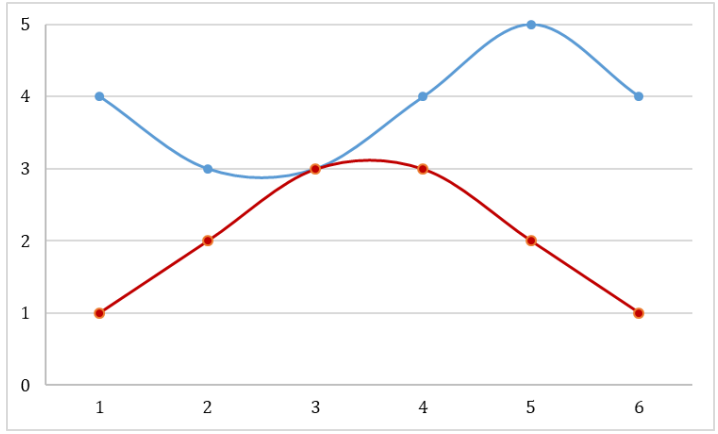
$$C = (2,3,3,3,2,2), Q = (1,2,2,3,3,2)$$

d	2	3	3	3	2	2	
2	0	1	1	1	0	0	6
3	1	0	0	0	1	1	5
3	1	0	0	0	1	1	4
2	0	1	1	1	0	0	3
2	0	1	1	1	0	0	2
1	1	4	4	4	1	1	1
→ j	1	2	3	4	5	6	↑ i



D	2	3	3	3	2	2	
2	+∞	3	2	2	2	1	**1**	6
3	+∞	3	1	1	1	2	3	5
3	+∞	2	1	1	1	2	3	4
2	+∞	1	2	3	4	4	4	3
2	+∞	1	2	3	4	4	4	2
1	+∞	1	5	9	13	14	15	1
	0	+∞	+∞	+∞	+∞	+∞	+∞	0
→ j	0	1	2	3	4	5	6	↑ i

Пример вычисления DTW



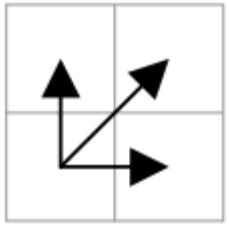
$$\mathrm{DTW}(Q, C) = D(m, m)$$

$$D(i,j) = (q_i - c_j)^2 + \min \begin{cases} D(i-1,j) \\ D(i,j-1) \\ D(i-1,j-1) \end{cases}$$

$$D(0,0) = 0, \qquad D(i,0) = D(0,j) = +\infty$$

$$C = (4,3,3,4,5,4), Q = (1,2,3,3,2,1)$$

d	4	3	3	4	5	4	
1	9	4	4	9	16	9	6
2	4	1	1	4	9	4	5
3	1	0	0	1	4	1	4
3	1	0	0	1	4	1	3
2	4	1	1	4	9	4	2
1	9	4	4	9	16	9	1
$\rightarrow j$	1	2	3	4	5	6	$\uparrow i$



D	4	3	3	4	5	4	
1	$+\infty$	28	15	15	20	30	**28**	6
2	$+\infty$	19	11	11	14	20	19	5
3	$+\infty$	15	10	10	11	15	16	4
3	$+\infty$	14	10	10	11	15	16	3
2	$+\infty$	13	10	11	15	24	28	2
1	$+\infty$	9	13	17	26	42	51	1
	0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	0
$\rightarrow j$	0	1	2	3	4	5	6	$\uparrow i$

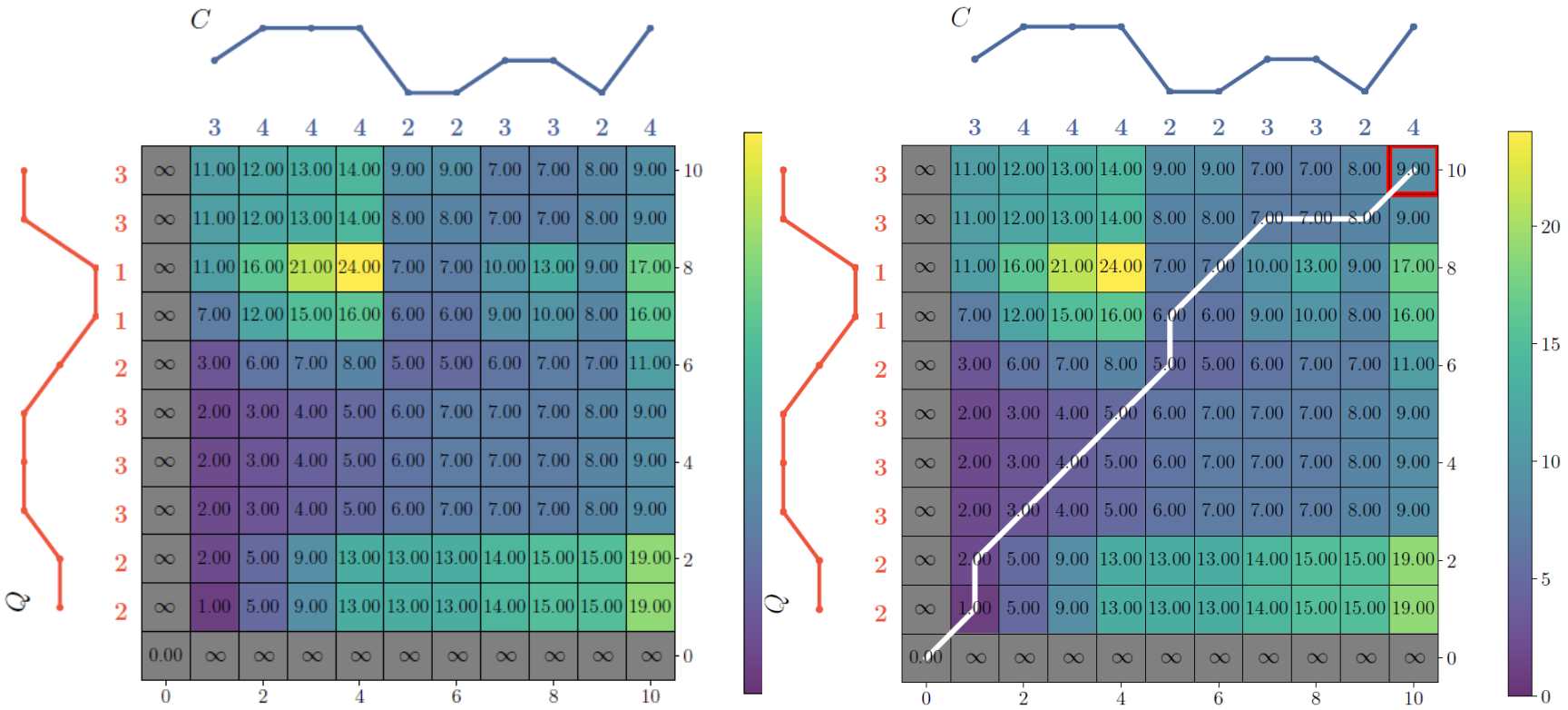
Построение оптимального пути трансформации

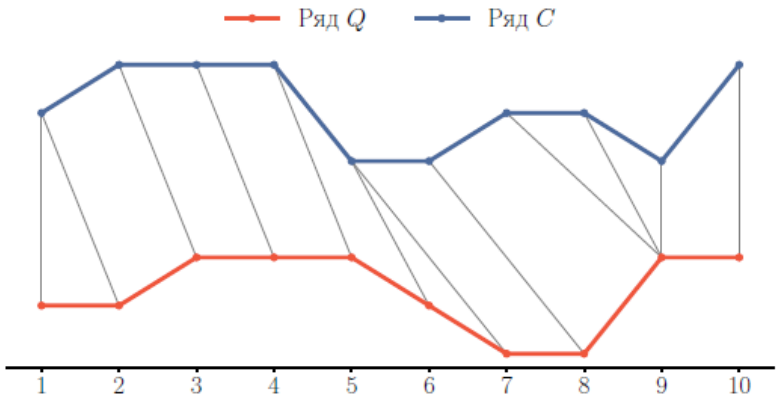
Вход: ряды Q, C длины m, матрица глобальной стоимости $D \in \mathbb{R}^{(1+m)\times(1+m)}$
Выход: оптимальный путь трансформации W^*

$i \leftarrow m$; $j \leftarrow m$; $k \leftarrow 0$; $W^* \leftarrow \varnothing$
while $i > 1$ **or** $j > 1$ **do**
  $k \leftarrow k + 1$; $w_k^* \leftarrow (i, j)$ ▷ Добавить текущую ячейку
  $diagCost \leftarrow +\infty$; $downCost \leftarrow +\infty$; $leftCost \leftarrow +\infty$
  ▷ Выбрать направление движения
  if $i > 1$ **and** $j > 1$ **then**
    $diagCost \leftarrow D(i-1, j-1)$
  end if
  if $i > 1$ **then**
    $downCost \leftarrow D(i-1, j)$
  end if
  if $j > 1$ **then**
    $leftCost \leftarrow D(i, j-1)$
  end if
  if $diagCost \leqslant downCost$ **and** $diagCost \leqslant leftCost$ **then**
    $i \leftarrow i - 1$; $j \leftarrow j - 1$
  else
    if $downCost \leqslant leftCost$ **then**
      $i \leftarrow i - 1$
    else
      $j \leftarrow j - 1$
    end if
  end if
end while
$k \leftarrow k + 1$; $w_k^* \leftarrow (1, 1)$ ▷ Добавить первую ячейку
$W^* \leftarrow \textsc{Reverse}(W^*)$ ▷ Обратить построенный путь
return W^*

Оптимальный путь восстанавливается из матрицы глобальной стоимости движением в порядке, обратном заполнению матрицы – от (m, m) к $(1, 1)$ – и затем обращается

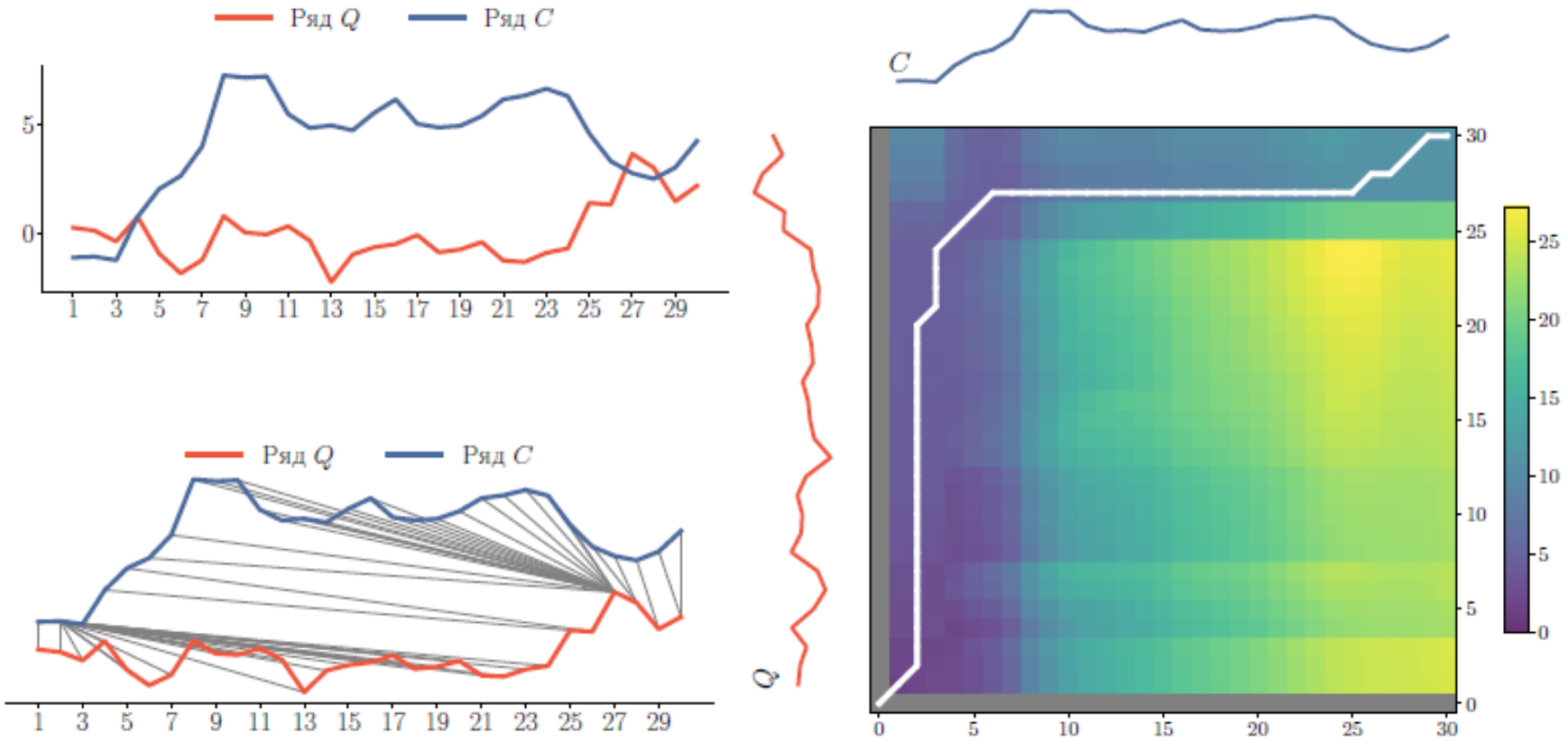
Построение оптимального пути





Патологическое выравнивание рядов

- DTW может полагать оптимальными интуитивно непонятные и нелогичные пути трансформации: вычисленное расстояние не будет отражать реальную схожесть

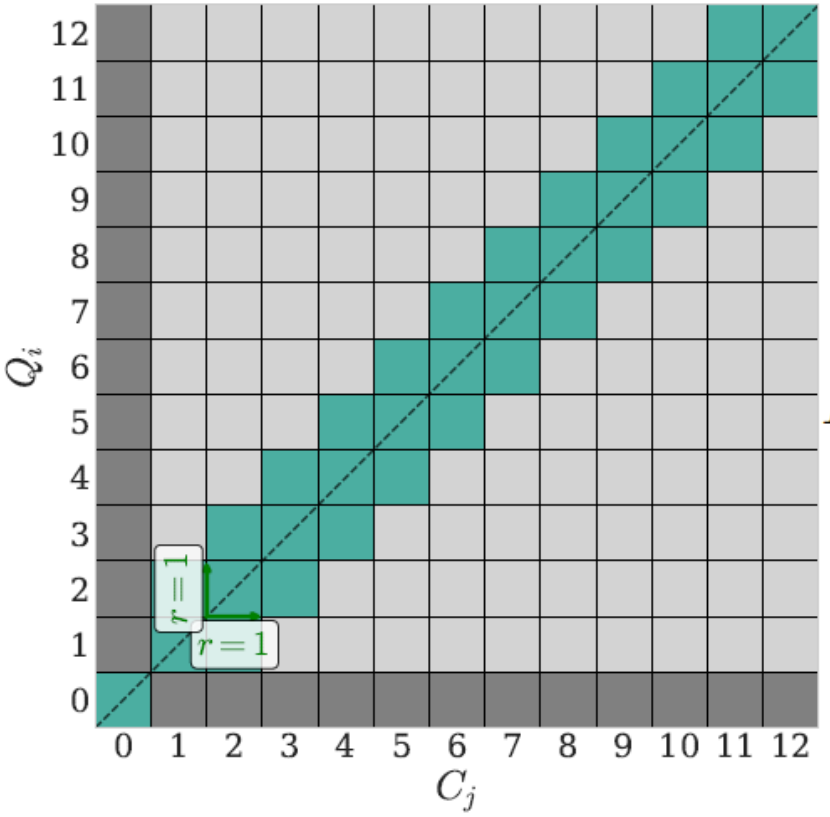


Предотвращение патологического выравнивания

- Полоса Сакоэ—Тибы
- Параллелограмм Итакуры
- Нормализация стоимости пути

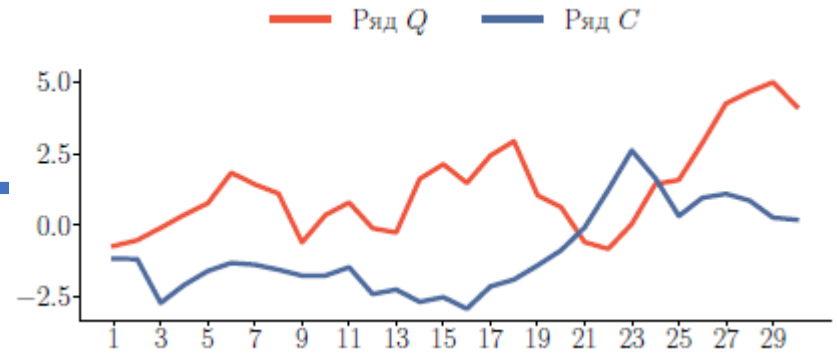
Полоса Сакоэ—Тибы

- Запрещает отклонение пути трансформации более чем на r ячеек $(0 \leq r \leq m-1)$ от главной диагонали матрицы глобальной стоимости

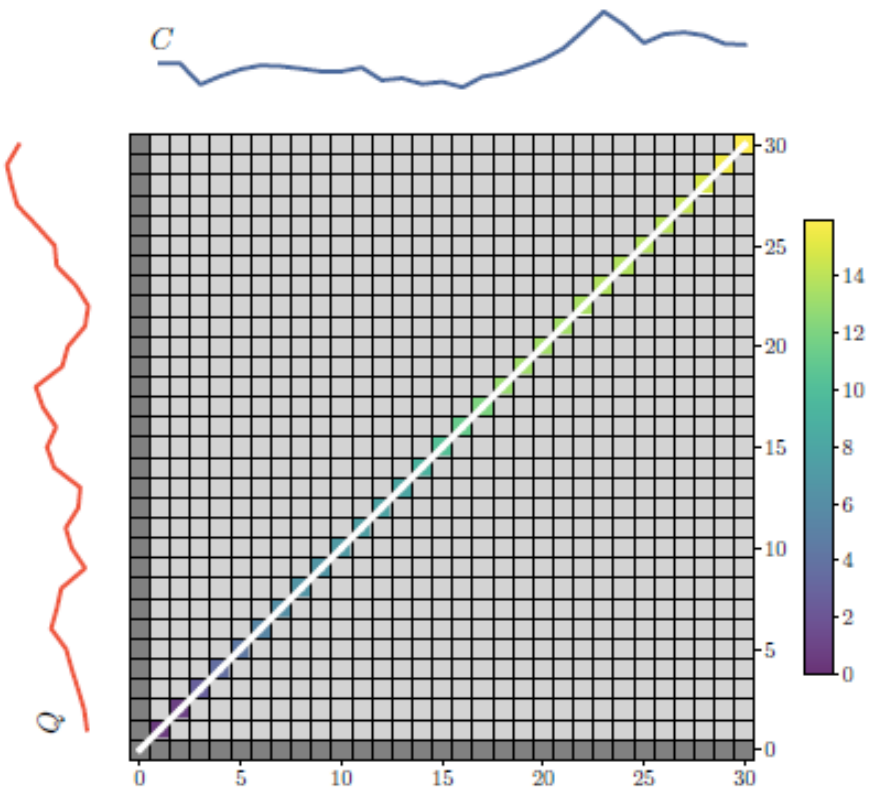


$$D(i,\,j) = \begin{cases} +\infty, & 1 \leqslant i \leqslant m, \\ & \min(m, i+r) < j < \max(1, i-r) \\ d(i,j) + \min\big(D(i-1, j-1), & \\ \qquad D(i-1, j), D(i, j-1)\big), & \text{иначе} \end{cases}$$

Вычисление DTW с полосой Сакоэ—Тибы

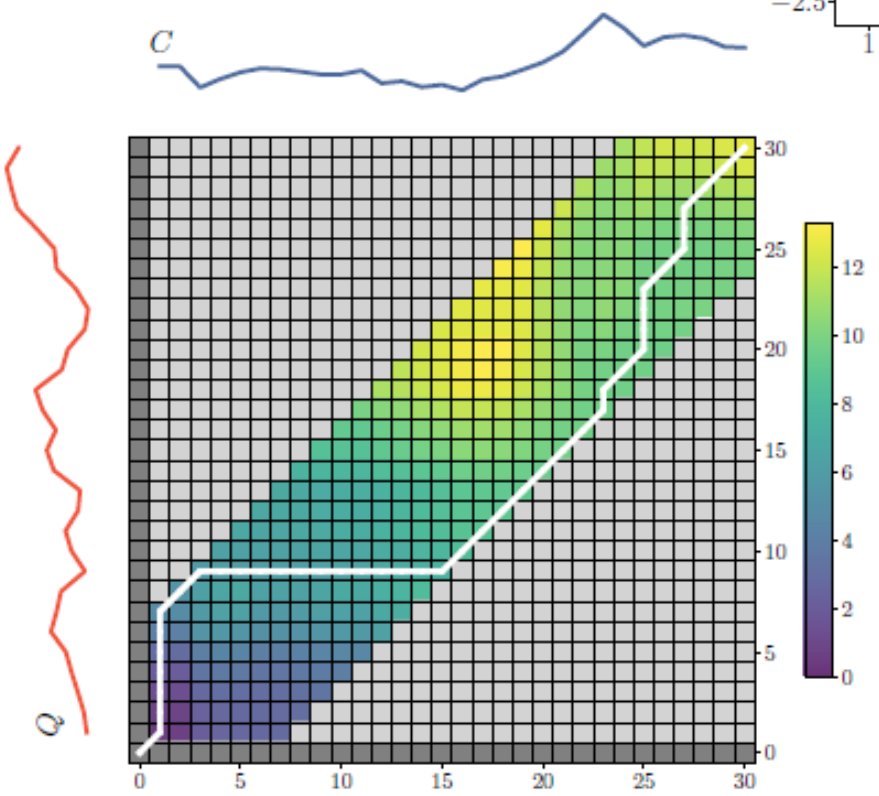


$r = 0$



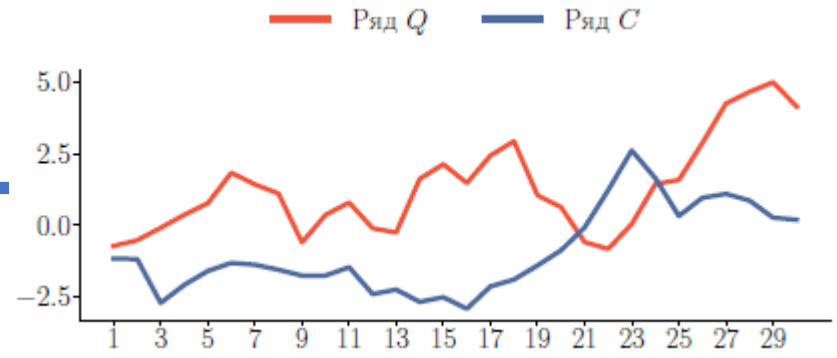
$\mathrm{DTW}(Q, C) = 15.961$

$r = 0.25m$

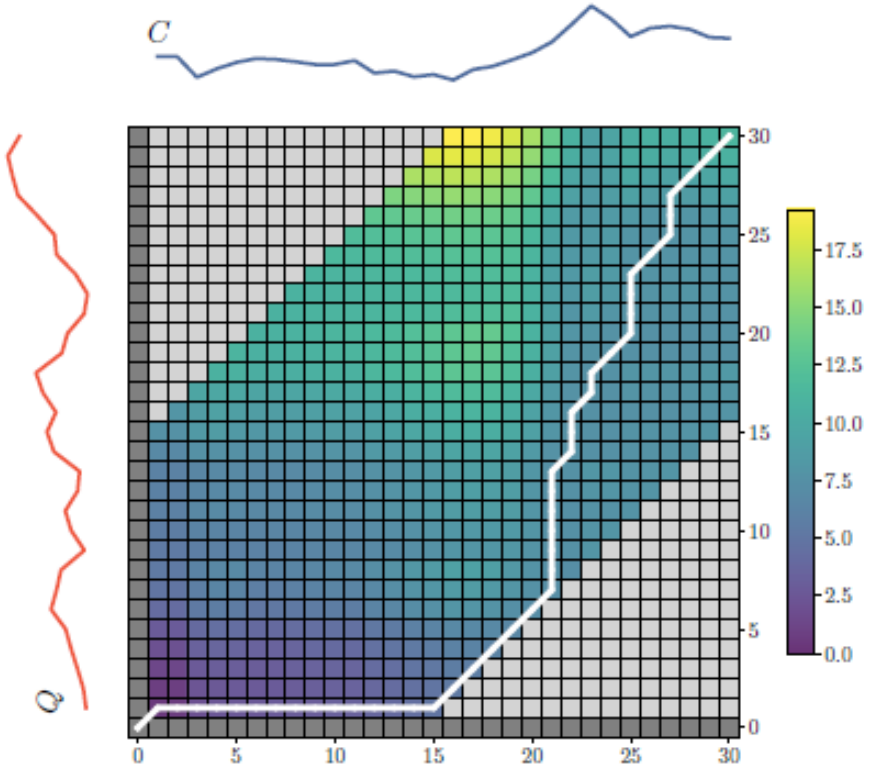


$\mathrm{DTW}(Q, C) = 12.644$

Вычисление DTW с полосой Сакоэ—Тибы

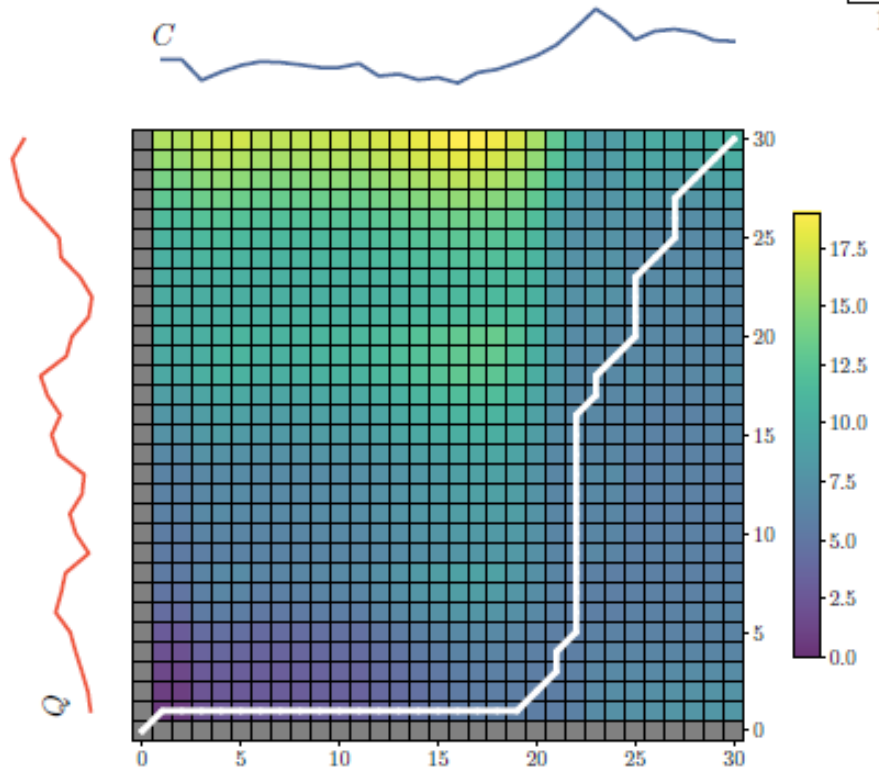


$r = 0.5m$



$\mathrm{DTW}(Q, C) = 11.182$

$r = m - 1$



$\mathrm{DTW}(Q, C) = 10.528$

Алгоритм cDTW (constraint DTW)

Вход: ряды Q, C длины m, ограничение r; $0 \leqslant r \leqslant m-1$
Выход: расстояние DTW от Q до C
Данные: матрица локальной стоимости $d \in \mathbb{R}^{m \times m}$,
матрица глобальной стоимости $D \in \mathbb{R}^{(1+m)\times(1+m)}$

1: $D(0, \cdot) \leftarrow +\infty_{1+m}$; $D(\cdot, 0) \leftarrow +\infty_{1+m}$; $D(0, 0) \leftarrow 0$
2: **for** $i \leftarrow 1$ **to** m **do**
3: **for** $j \leftarrow \max(1, i-r)$ **to** $\min(m, i+r)$ **do**
4: $d(i, j) \leftarrow \mathrm{Cost}(q_i, c_j)$
5: $D(i, j) \leftarrow d(i, j) + \min\big(D(i-1, j-1),\ D(i-1, j),\ D(i, j-1)\big)$
6: **end for**
7: **end for**
8: **return** $\sqrt{D(m, m)}$

Алгоритм cDTW (constraint DTW)

Вход: ряды Q, C длины m, ограничение r; $0 \leqslant r \leqslant m-1$

Выход: расстояние DTW от Q до C

Данные: матрица локальной стоимости $d \in \mathbb{R}^{m \times m}$,
матрица глобальной стоимости $D \in \mathbb{R}^{(1+m)\times(1+m)}$

Пространственная сложность $\boldsymbol{O(m^2)}$

1: $D(0, \cdot) \leftarrow +\infty_{1+m}$; $D(\cdot, 0) \leftarrow +\infty_{1+m}$; $D(0, 0) \leftarrow 0$
2: **for** $i \leftarrow 1$ **to** m **do**
3: **for** $j \leftarrow \max(1, i-r)$ **to** $\min(m, i+r)$ **do**
4: $d(i, j) \leftarrow \mathrm{Cost}(q_i, c_j)$
5: $D(i, j) \leftarrow d(i, j) + \min\big(D(i-1, j-1),\ D(i-1, j),\ D(i, j-1)\big)$
6: **end for**
7: **end for**
8: **return** $\sqrt{D(m, m)}$

Вычислительная сложность $\boldsymbol{O(rm)}$

Алгоритм cDTW (constraint DTW)

Вход: ряды Q, C длины m, ограничение r; $0 \leqslant r \leqslant m-1$
Выход: расстояние DTW от Q до C
Данные: матрица локальной стоимости $d \in \mathbb{R}^{m \times m}$,
матрица глобальной стоимости $D \in \mathbb{R}^{(1+m)\times(1+m)}$

1: $D(0, \cdot) \leftarrow +\infty_{1+m}$; $D(\cdot, 0) \leftarrow +\infty_{1+m}$; $D(0, 0) \leftarrow 0$
2: **for** $i \leftarrow 1$ **to** m **do**
3: **for** $j \leftarrow \max(1, i-r)$ **to** $\min(m, i+r)$ **do**
4: $d(i, j) \leftarrow \mathrm{Cost}(q_i, c_j)$
5: $D(i, j) \leftarrow d(i, j) + \min\big(D(i-1, j-1), D(i-1, j), D(i, j-1)\big)$
6: **end for**
7: **end for**
8: **return** $\sqrt{D(m, m)}$

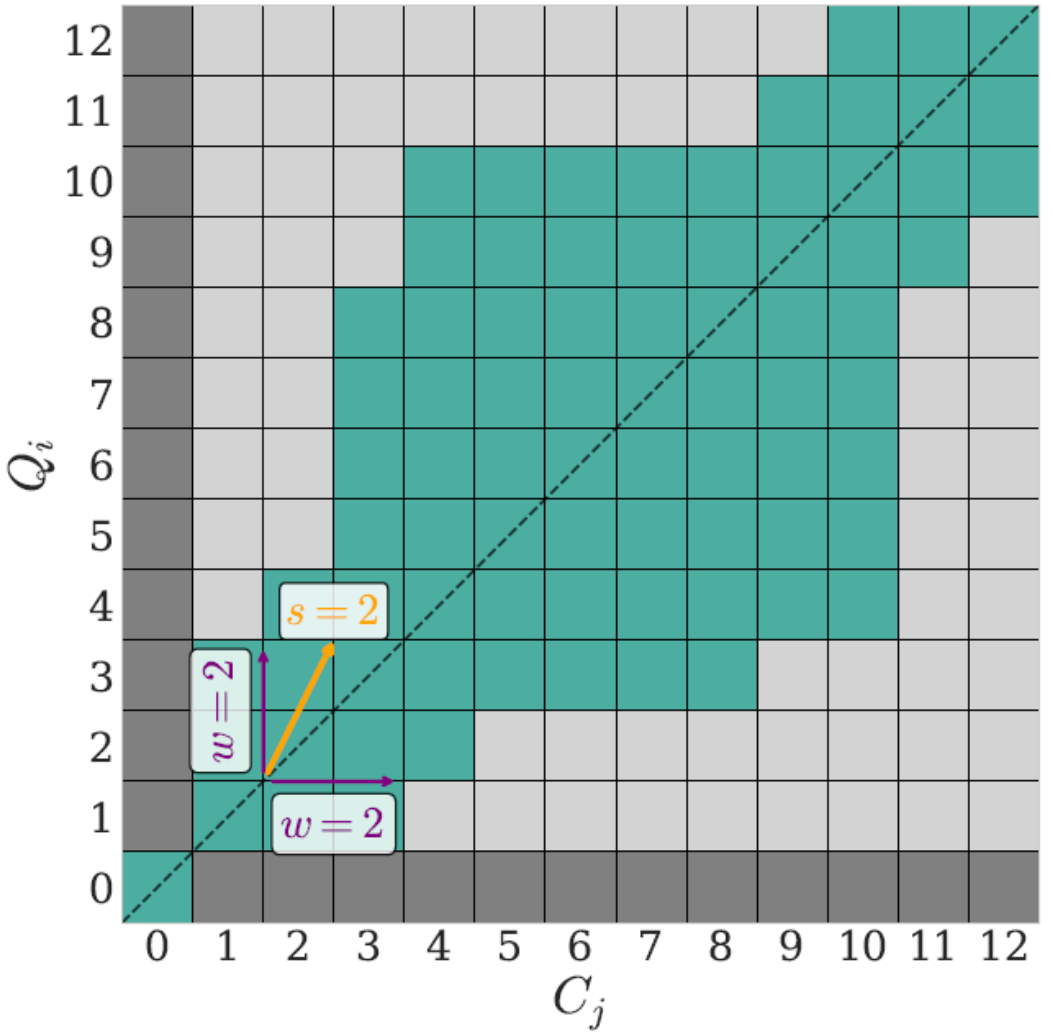
Параметр r подбирается экспериментально, нач. значение – 5-10% от m

Немного выравнивания – это хорошо, но слишком много выравнивания – это плохо*

* Ratanamahatana C.A., Keogh E.J. Three myths about Dynamic Time Warping data mining. SDM 2005. https://doi.org/10.1137/1.9781611972757.50

Параллелограмм Итакуры

- Формируется вокруг главной диагонали матрицы глобальной стоимости
  - наклон s $(1 \leq s < m)$ задает максимальное отклонение пути трансформации от диагонали по оси i или j на единицу изменения по другой оси
  - окно w $(0 \leq w < m)$ задает, насколько далеко в начале или конце может отходить путь трансформации от диагонали11

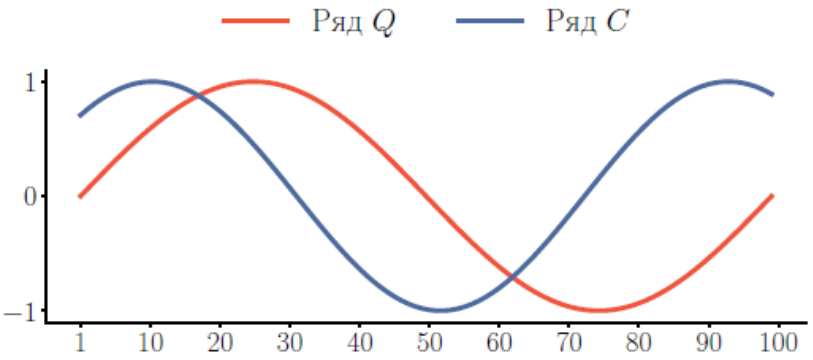


Нормализация стоимости пути

- Формула $\mathrm{Cost}_W(Q,C) = \sqrt{\sum_{k=1}^{K} d(w_k)}$ дает зависимость стоимости пути от его длины: более длинные пути стоят больше, но отвергаются, хотя могут дать лучшее выравнивание
- Формула с нормализацией стоимости относительно длины пути («средняя стоимость выравнивания рядов» на одну точку):

$$\mathrm{Cost}_W(Q,C) = \frac{1}{K}\sqrt{\sum_{k=1}^{K} d(w_k)}$$

Пример выгоды от нормализации стоимости пути



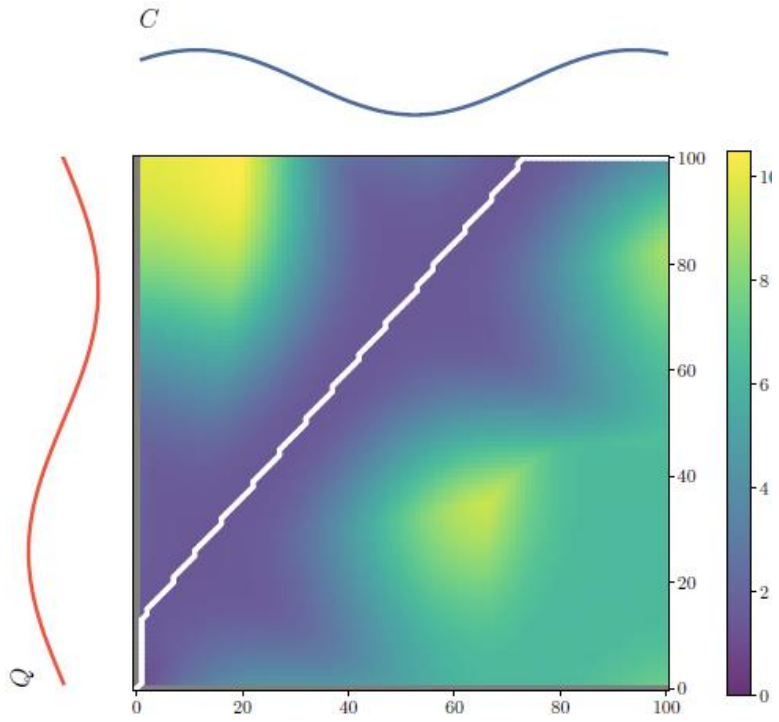
$$\mathrm{Cost}_W(Q,C) = \sqrt{\sum_{k=1}^{K} d(w_k)}$$

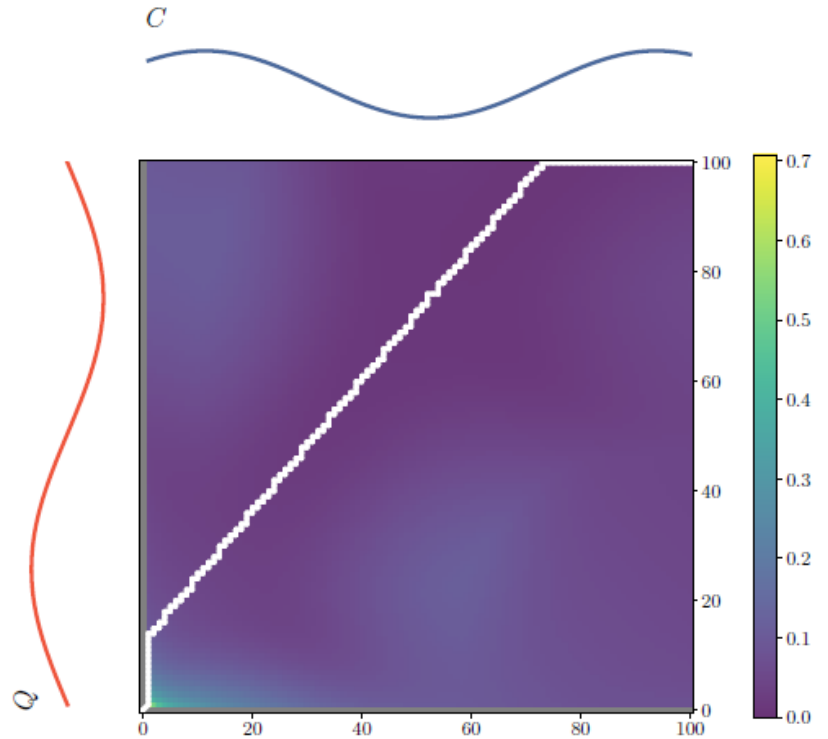
$$\mathrm{DTW}(Q,C) = \mathbf{4.318}, |W^*| = \mathbf{128}$$

$$\mathrm{Cost}_W(Q,C) = \frac{1}{K}\sqrt{\sum_{k=1}^{K} d(w_k)}$$

$$\mathrm{DTW}(Q,C) = 0.022, |W^*| = \mathbf{199}$$

$$m = 100$$
$$q_i = \sin t_i$$
$$c_i = \sin(1.2t_i + \pi/4)$$
$$t_i = 2\pi/99$$





Литература

Цымблер М.Л. Интеллектуальный анализ временных рядов: учебное пособие. Челябинск: Издательский центр ЮУрГУ, 2026. 315 с. (в печати)

Глава 2. Функции расстояния

https://mzym.susu.ru/courses/tsm/tsmbook.pdf