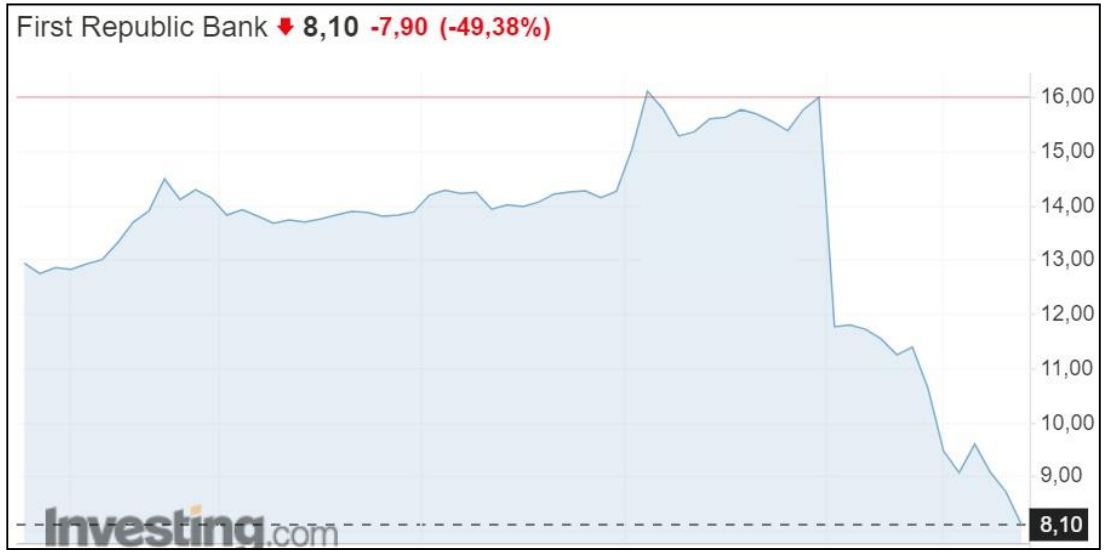
Поиск подпоследовательностей временного ряда по образцу

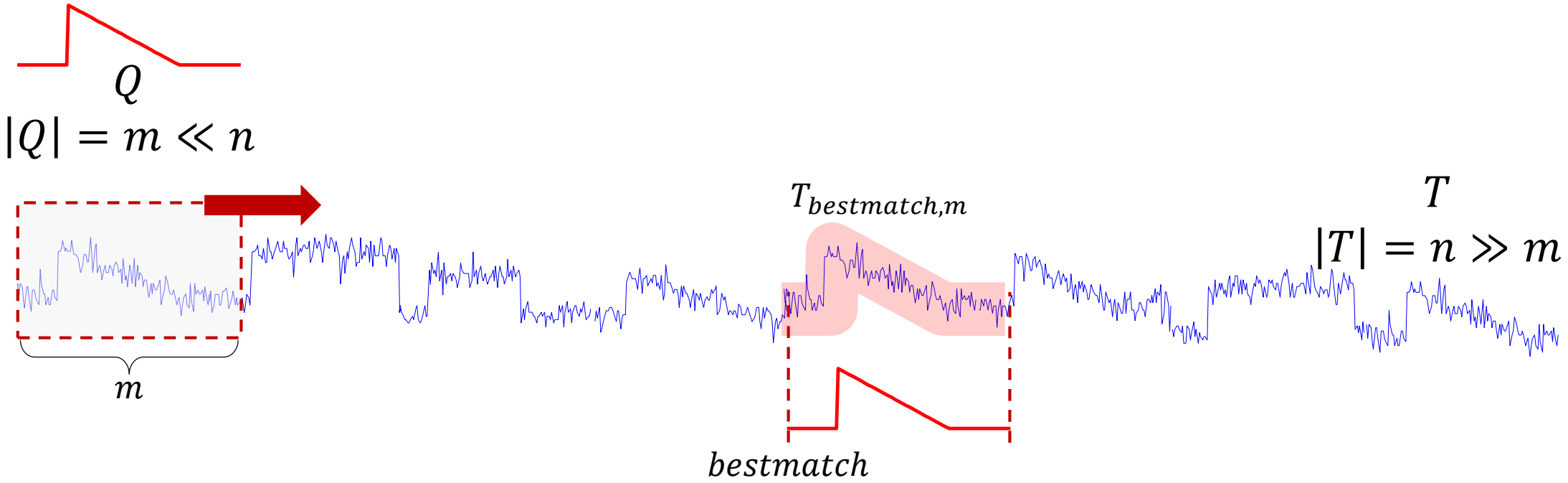


Возьми себе в образец героя древних времен, наблюдай его, иди за ним вслед, поравняйся, обгони – слава тебе!

А.В. Суворов

Содержание

Поиск по образцу (subsequence matching/similarity search)



В ряде T найти подпоследовательность $T_{bestmatch,m}$, наиболее похожую на запрос Q:

$$\forall T_{i,m} \in S_T^m \;\; \mathrm{Dist}\left(T_{bestmatch,m}, Q\right) \leq \mathrm{Dist}\left(T_{i,m}, Q\right)$$

Какую взять функцию Dist(·,·) для измерения схожести?

- Евклидово расстояние
- DTW (Dynamic Time Warping, динамическая трансформация времени)

- Расстояние Махалонобиса
- Расстояние Хэмминга
- Расстояние Левенштейна
- Косинусная мера схожести
- …

Содержание

Не-метрика (мера) расстояния $\mathrm{Dist}: M \times M \to \mathbb{R}$**: выполнены** $\forall x, y, z \in M$

- Аксиома тождества:

$$\mathrm{Dist}(x, x) = 0$$

- Аксиома положительности:

$$\mathrm{Dist}(x, y) \geq 0$$

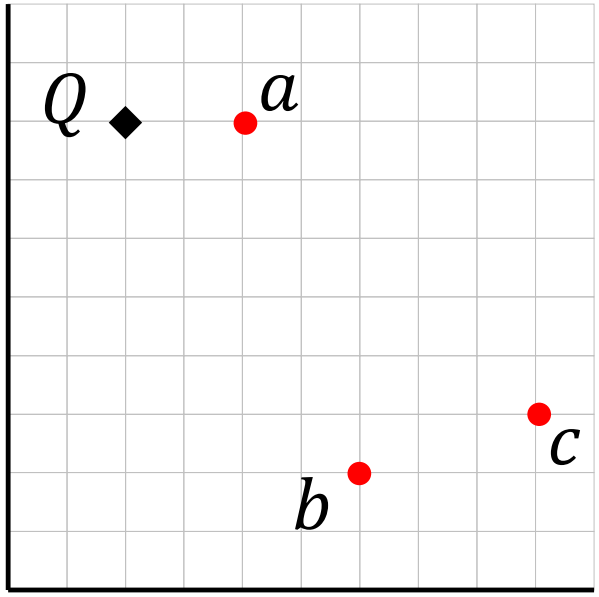
- Аксиома симметричности:

$$\mathrm{Dist}(x, y) = \mathrm{Dist}(y, x)$$

- **НО не выполнена аксиома (неравенство) треугольника**:

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

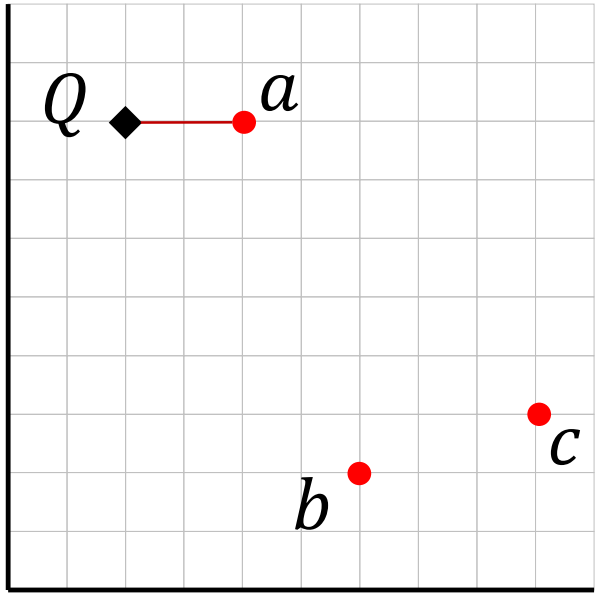
Почему важно неравенство треугольника



- Поиск в множестве $T = \{a, b, c, \ldots\}$ объекта, ближайшего к Q:

Dist(·,·)	a	b	c
a	0	6.32	7.07
b	6.32	0	3.16
c	7.07	3.16	0

Почему важно неравенство треугольника

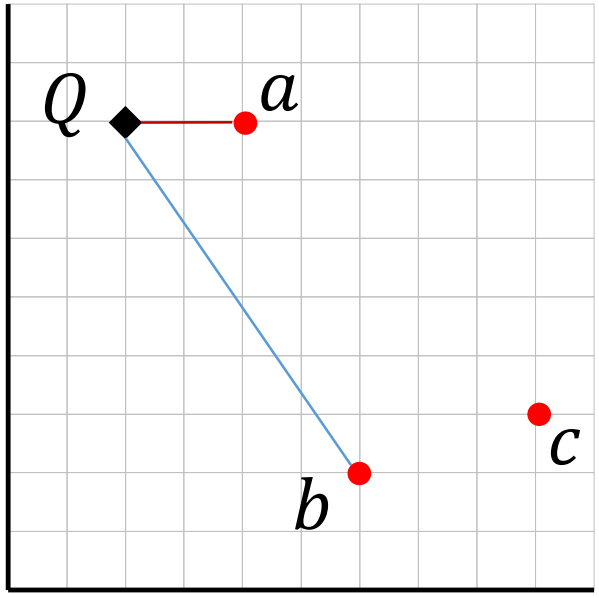


- Поиск в множестве $T = \{a, b, c, \ldots\}$ объекта, ближайшего к Q:

1. $\mathrm{Dist}(Q, a) = 2$ (***bsf***, ***best-so-far***)

Dist(·,·)	a	b	c
a	0	6.32	7.07
b	6.32	0	3.16
c	7.07	3.16	0

Почему важно неравенство треугольника

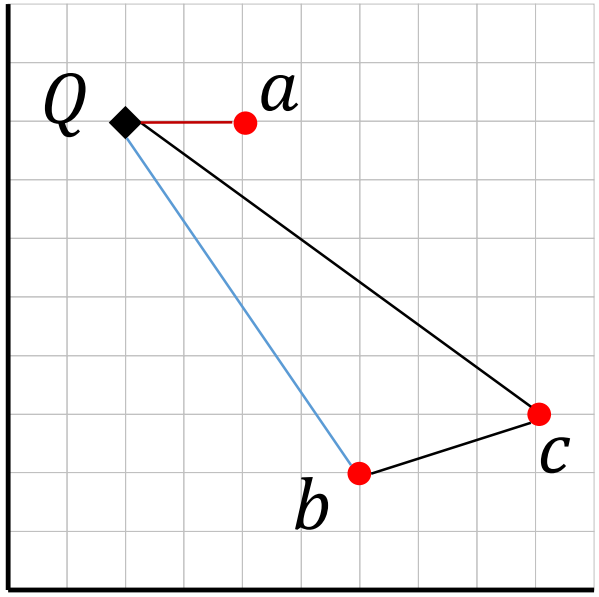


- Поиск в множестве $T = \{a, b, c, \ldots\}$ объекта, ближайшего к Q:

1. $\mathrm{Dist}(Q, a) = 2$ (***bsf***, ***best-so-far***)
2. $\mathrm{Dist}(Q, b) = 7.21$

Dist(·,·)	a	b	c
a	0	6.32	7.07
b	6.32	0	3.16
c	7.07	3.16	0

Почему важно неравенство треугольника

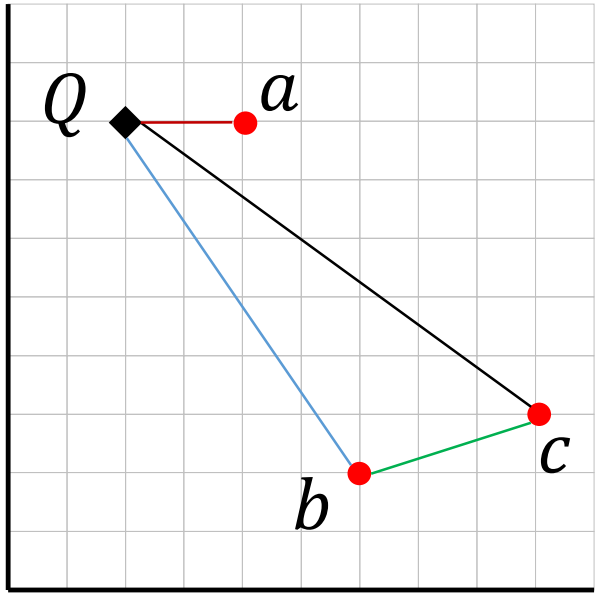


Dist(·,·)	a	b	c
a	0	6.32	7.07
b	6.32	0	3.16
c	7.07	3.16	0

- Поиск в множестве $T = \{a, b, c, \dots\}$ объекта, ближайшего к Q:

1. $\mathrm{Dist}(Q, a) = 2$ (***bsf***, ***best-so-far***)
2. $\mathrm{Dist}(Q, b) = 7.21$
3. $\triangle Qbc$:
   $\mathrm{Dist}(Q, b) \leq \mathrm{Dist}(Q, c) + \mathrm{Dist}(b, c)$

Почему важно неравенство треугольника



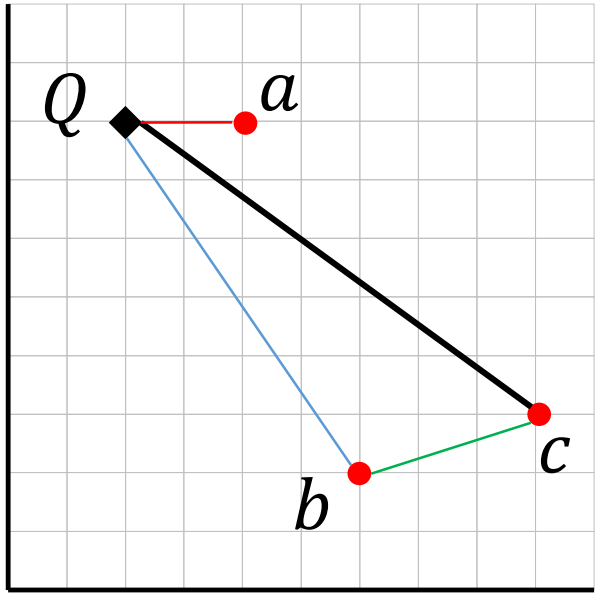
Dist(·,·)	a	b	c
a	0	6.32	7.07
b	6.32	0	3.16
c	7.07	3.16	0

- Поиск в множестве $T = \{a, b, c, \dots\}$ объекта, ближайшего к Q:

1. $\mathrm{Dist}(Q, a) = 2$ (***bsf***, ***best-so-far***)
2. $\mathrm{Dist}(Q, b) = 7.21$
3. $\triangle Qbc$:

$$\mathrm{Dist}(Q, b) \le \mathrm{Dist}(Q, c) + \mathrm{Dist}(b, c)$$
$$\mathrm{Dist}(Q, b) - \mathrm{Dist}(b, c) \le \mathrm{Dist}(Q, c)$$
$$7.21 - 3.16 \le \mathrm{Dist}(Q, c)$$
$$4.05 \le \mathrm{Dist}(Q, c)$$

Почему важно неравенство треугольника



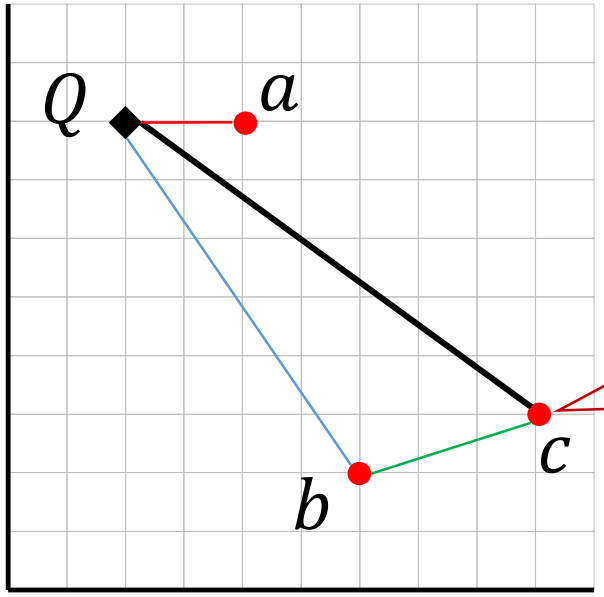
Dist(·,·)	a	b	c
a	0	6.32	7.07
b	6.32	0	3.16
c	7.07	3.16	0

- Поиск в множестве $T = \{a, b, c, \dots\}$ объекта, ближайшего к Q:

1. $\mathrm{Dist}(Q, a) = 2$ (***bsf***, ***best-so-far***)
2. $\mathrm{Dist}(Q, b) = 7.21$
3. $\triangle Qbc$:
   $\mathrm{Dist}(Q, b) \leq \mathbf{Dist}(Q, c) + \mathrm{Dist}(b, c)$
   $\mathrm{Dist}(Q, b) - \mathrm{Dist}(b, c) \leq \mathbf{Dist}(Q, c)$
   $7.21 - 3.16 \leq \mathbf{Dist}(Q, c)$
   $4.05 \leq \mathbf{Dist}(Q, c)$
   $2 = \mathrm{Dist}(Q, a) < 4.05 \leq \mathbf{Dist}(Q, c)$

- Объект c **можно отбросить без вычисления** $\mathbf{Dist}(Q, c)$

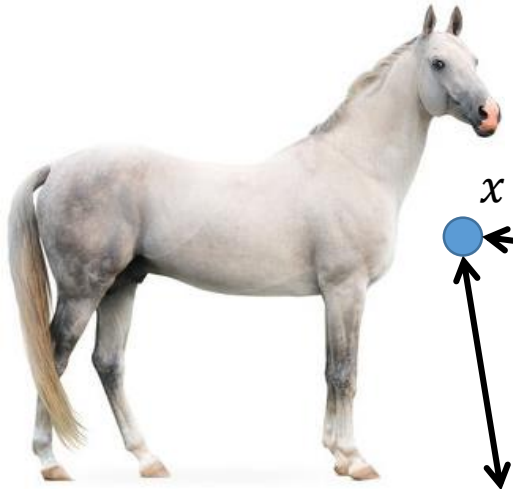
Почему важно неравенство треугольника

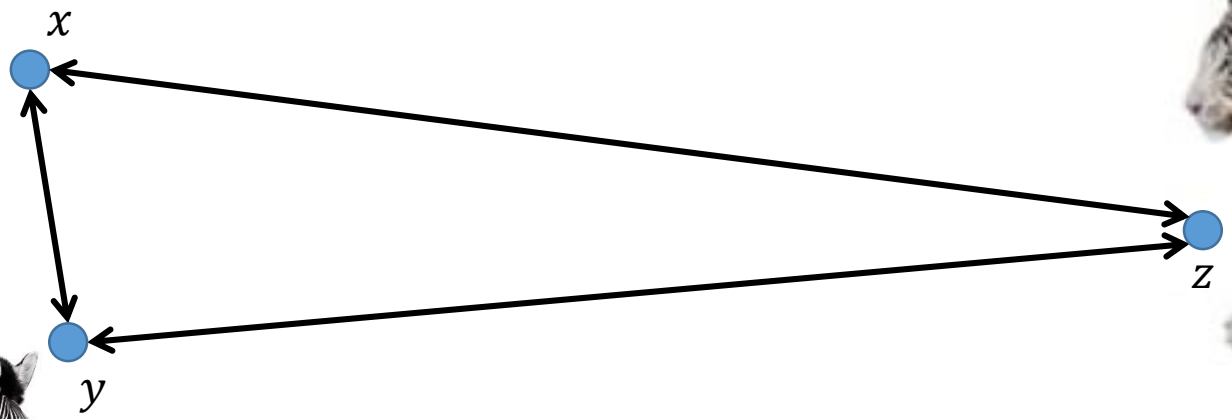


Dist(·,·)	a	b	c
a	0	6.32	7.07
b	6.32	0	3.16
c	7.07	3.16	0

При поиске по образцу с применением метрики неравенство треугольника дает возможность указать **нижнюю границу схожести**, чтобы **отбрасывать заведомо непохожие объекты без вычисления расстояния до них**

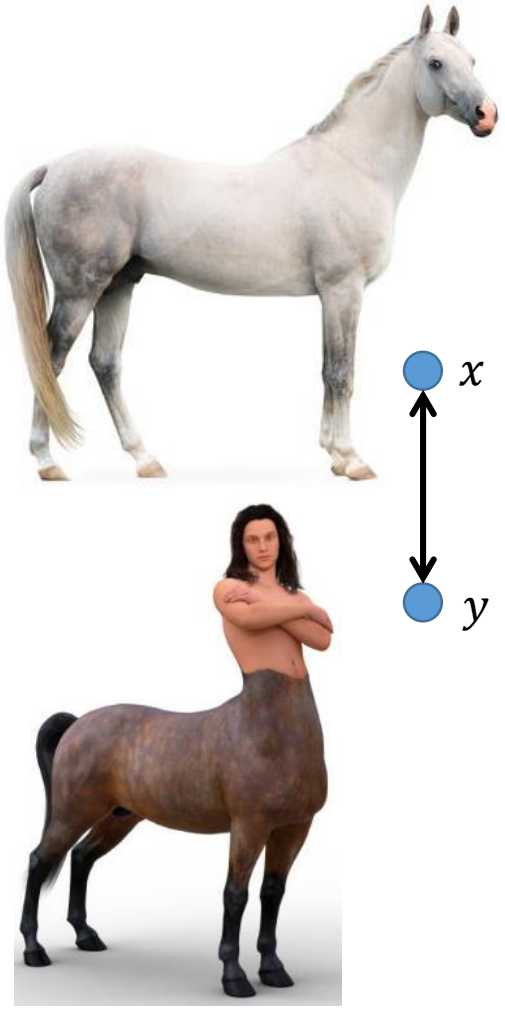
Метрика: неравенство треугольника

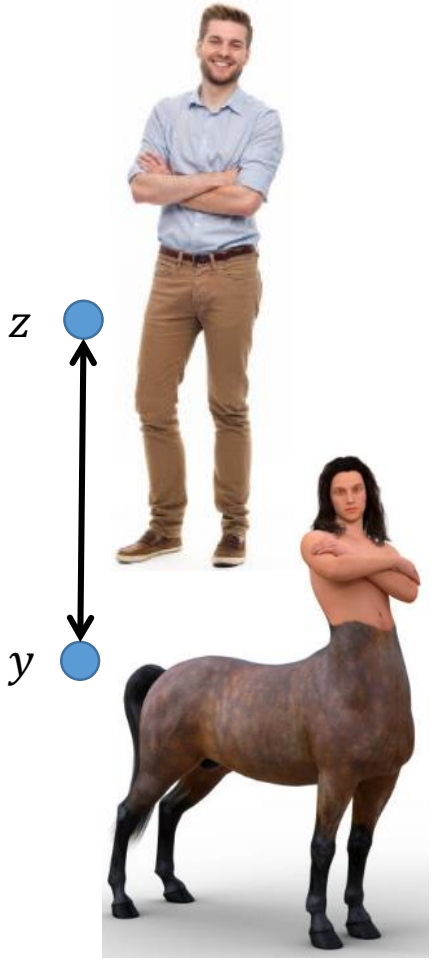




$$\forall x, y, z$$
$$\mathrm{Dist}(x, z) \leq \mathrm{Dist}(x, y) + \mathrm{Dist}(y, z)$$

Не-метрика: неравенство треугольника РАБОТАЕТ, НО НЕ ВСЕГДА





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

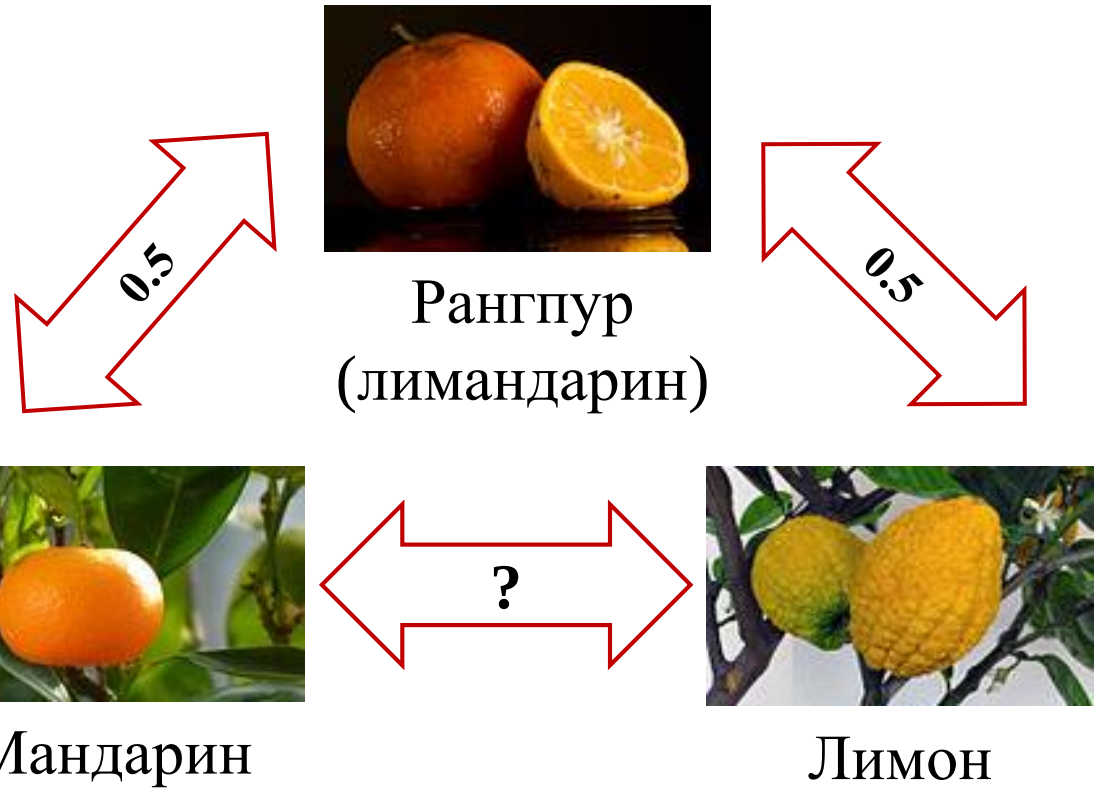
$$\Downarrow$$

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

Метрика и не-метрика без неравенства треугольника

ED, ED_{norm} и др.

ED^2, DTW, MPdist и др.

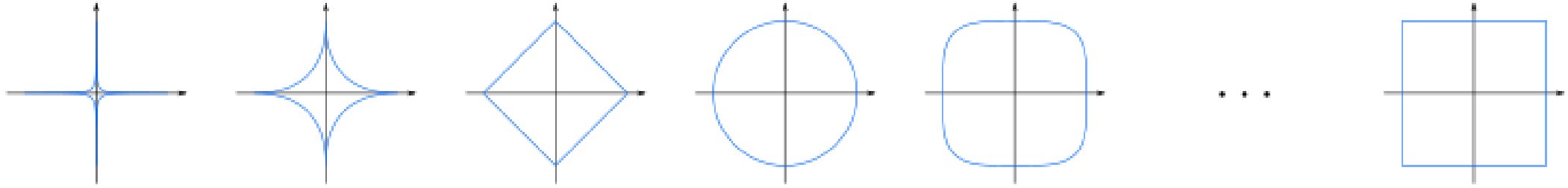


Расстояние Минковского

- $\mathrm{Dist}(Q, C) = \sqrt[p]{\sum_{i=1}^{m} |q_i - c_i|^p}$
- $p < 1$: не метрика (нет аксиомы треугольника)
- $p \geq 1$: метрика
  - $p = 1$: Манхэттенское расстояние
  - $p = 2$: Евклидово расстояние
  - $p = \infty$: расстояние Чебышёва

Герман Минковский
1864-1909

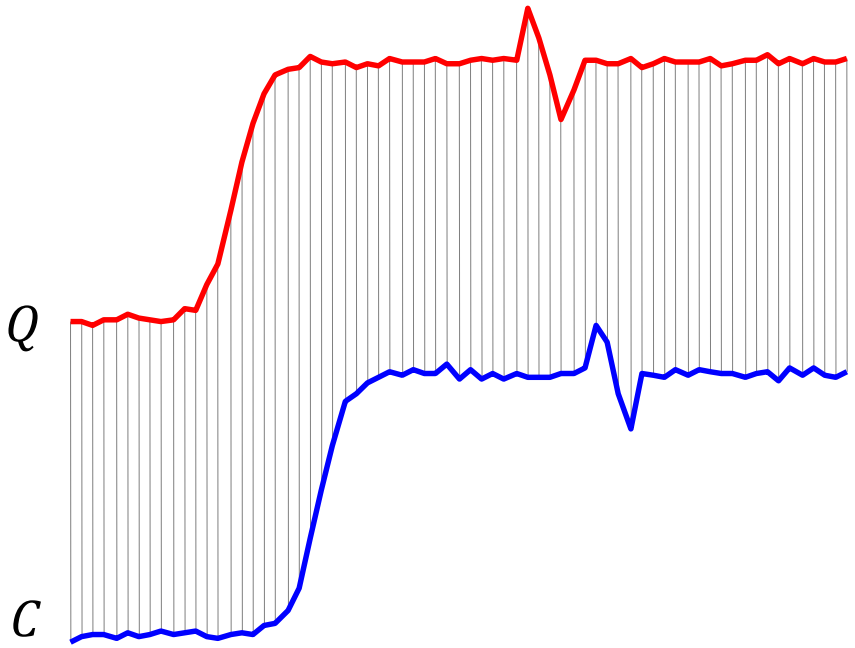
Единичная окружность при различных p

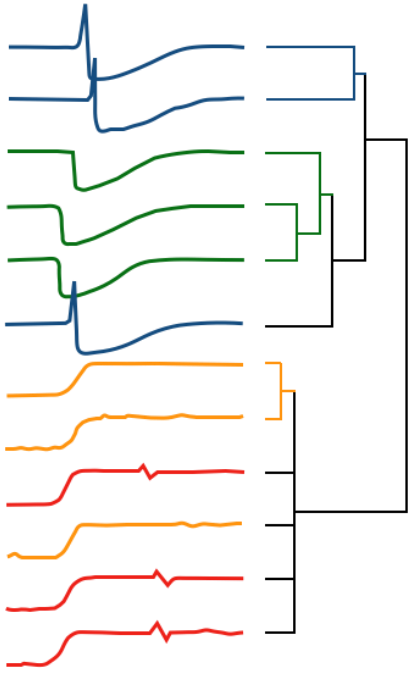


Содержание

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

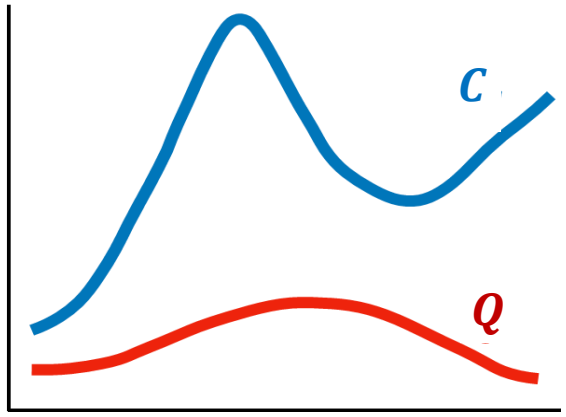
- Интуитивно понятное расстояние, вычислительная сложность $O(m)$
- NB! $\mathrm{ED}^2(Q,C)=\sum_{i=1}^{m}(q_i-c_i)^2$ вычисляется быстрее, но не метрика
- Сравнение равных по длине рядов по принципу «один к одному», не всегда адекватно учитывает форму рядов

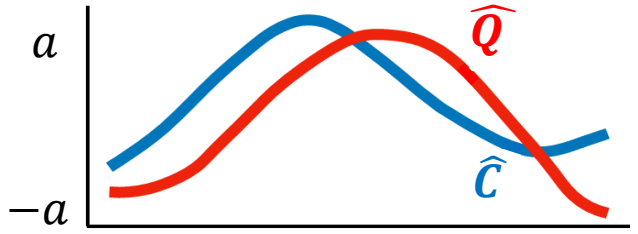




Евклид* 325-265 д.н.э.

Z-нормализация: сравнение рядов без учета разницы амплитуд





$$\mu_{\hat{Q}}=\mu_{\hat{C}}=0$$

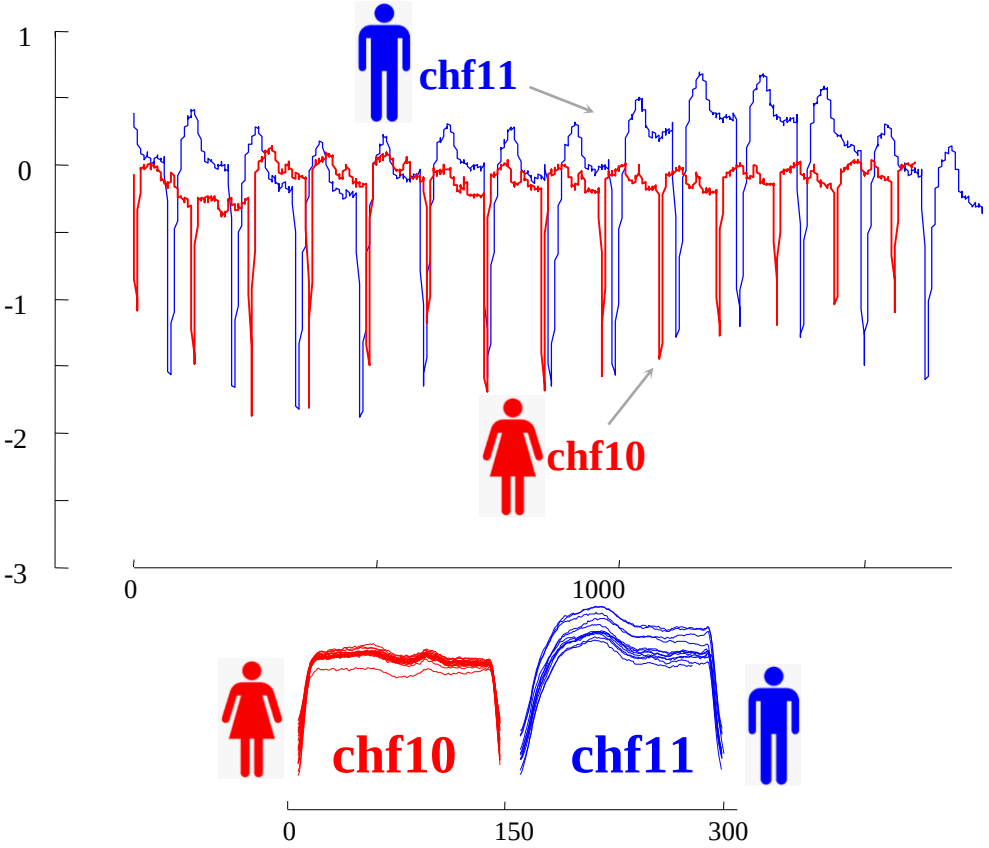
$$\sigma_{\hat{Q}}=\sigma_{\hat{C}}=1$$

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

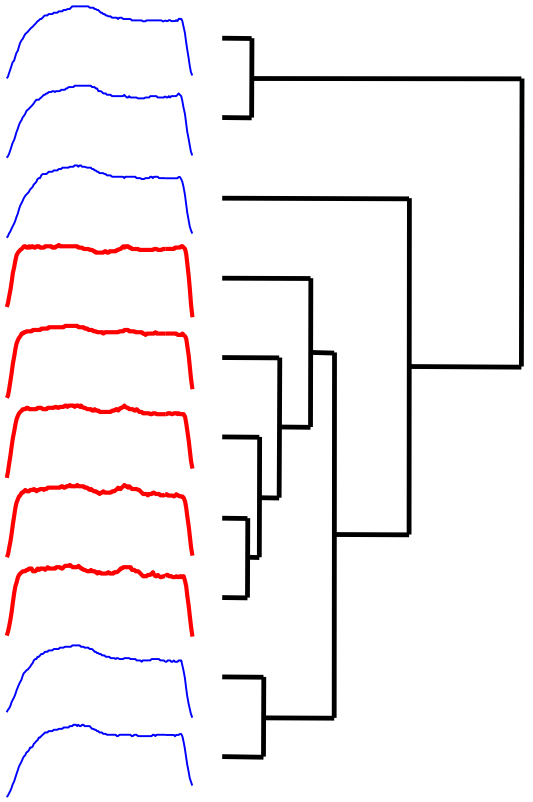
$$\mu=\frac{1}{n}\sum_{i=1}^{m}t_i,\qquad \sigma=\sqrt{\left(\frac{1}{m}\sum_{i=1}^{m}t_i^2\right)-\mu^2}$$

Важность z-нормализации

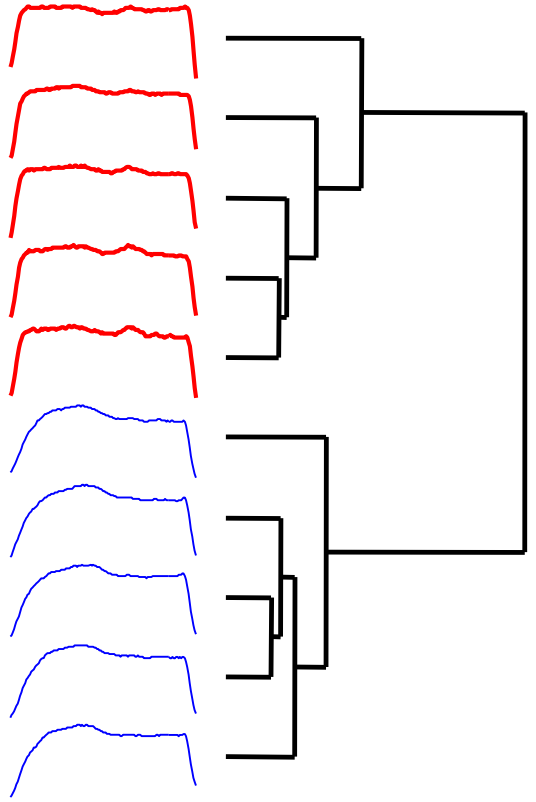
BIDMC Congestive Heart Failure Database*



Не нормализованные



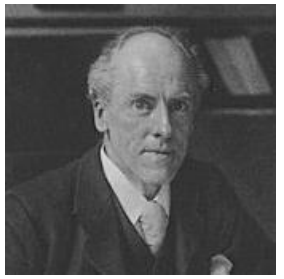
Z-нормализованные



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

Корреляция Пирсона как мера схожести

- $\mathrm{corr}(C,Q) = \frac{(\mathrm{M}_C - \mu_C)(\mathrm{M}_Q - \mu_Q)}{\sigma_C \sigma_Q} = \frac{\sum_{i=1}^{m} c_i q_i - m\mu_C\mu_Q}{m\sigma_C\sigma_Q} = \frac{\langle C,Q \rangle - m\mu_C\mu_Q}{m\sigma_C\sigma_Q}$
- $\mathrm{corr}(\cdot,\cdot) \in [-1,1]$ и потому не является метрикой
- *Достаточная статистика*
  - $\sum_{i=1}^{m} c_i q_i \quad \sum_{i=1}^{m} c_i \quad \sum_{i=1}^{m} q_i \quad \sum_{i=1}^{m} c_i^2 \quad \sum_{i=1}^{m} q_i^2$
  - Может быть вычислена за один проход
  - При наличии вычисленной достаточной статистики $\mathrm{corr}(\cdot,\cdot)$ вычисляется за $O(1)$



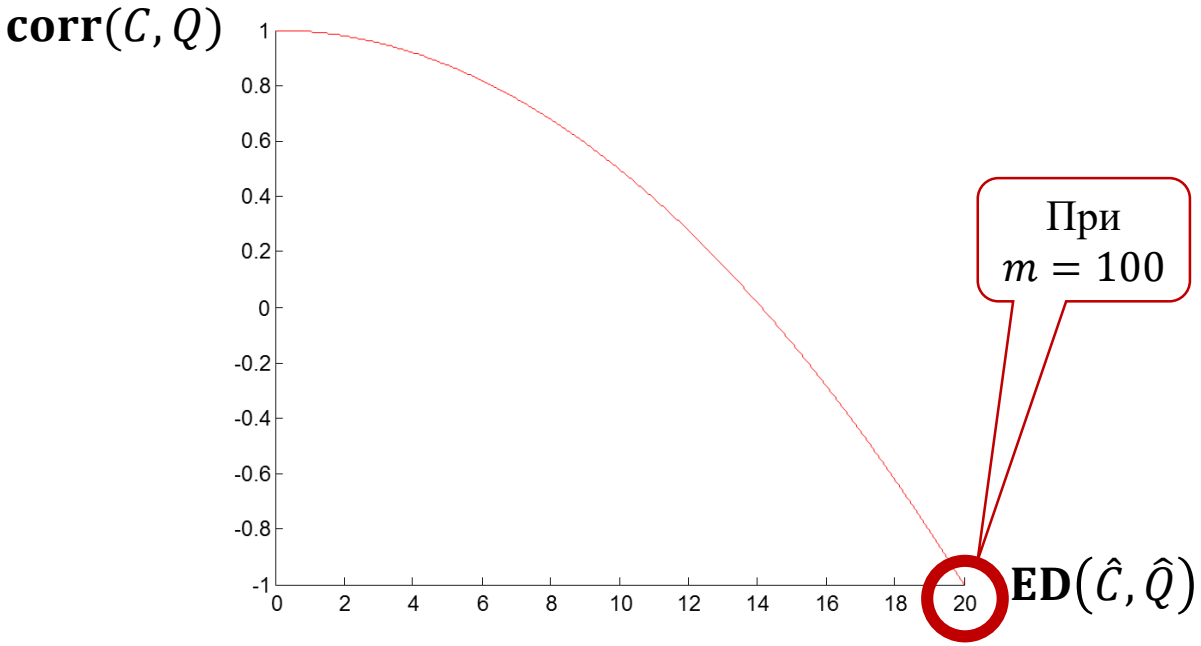
Карл Пирсон
(Karl Pearson)
1857-1936

Шкала Чеддока силы корреляции

+1.0	идеальная положительная
+0.9	очень сильная положительная
+0.7	сильная положительная
+0.4	умеренная положительная
+0.1	слабая положительная
0.0	отсутствие корреляции
−0.1	слабая отрицательная
−0.4	умеренная отрицательная
−0.7	сильная отрицательная
−0.9	очень сильная отрицательная
−1.0	идеальная отрицательная

Z-нормализация связана с корреляцией Пирсона*

- $$\mathrm{ED}(\hat{C},\hat{Q}) = \sqrt{2m(1-\mathrm{corr}(C,Q))} = \sqrt{2m(1-\frac{\langle C,Q\rangle - m\mu_C\mu_Q}{m\sigma_C\sigma_Q})}$$



* Mueen A., Nath S., Liu J. Fast approximate correlation for massive time-series data. Proc. of the ACM SIGMOD Int. Conf. on Management of Data, SIGMOD 2010, Indianapolis, Indiana, USA, June 6-10, 2010. P. 171–182. DOI: 10.1145/1807167.1807188

Содержание

- Постановка задачи
- Метрика и мера расстояния
- Расстояние Евклида
- **Алгоритм MASS**
- Мера DTW
- Поиск по образцу на основе DTW

Алгоритм MASS (Mueen's Algorithm for Similarity Search)

- MASS* – в настоящее время наиболее быстрый алгоритм вычисления *z-нормализованных евклидовых расстояний* от подпоследовательности-запроса до каждой подпоследовательности временного ряда
- Применение MASS
  - Поиск шаблонов в ряде
  - Вычисление матричного профиля и поиск мотивов ряда
  - Классификация и кластеризация рядов
  - Предсказание значений ряда
  - Поиск аномалий в ряде

Абдулла Муин (Abdullah Mueen)
Проф. Университета Нью-Мексико, США

* Zhong S., Mueen A. MASS: distance profile of a query over a time series. Data Min. Knowl. Discov. 2024. Vol. 38. P. 1466–1492. DOI: 10.1007/s10618-024-01005-2.

Алгоритм грубой силы (Brute Force)

Algorithm $BruteForce$

Input: запрос $Q \in \mathbb{R}^m$, ряд T

Output: профиль расстояния $D \in \mathbb{R}^{n-m+1}$, $D(i) = \mathrm{ED}(\hat{Q}, \hat{T}_{i,m})$

$D := \bar{0}$; $\hat{Q} := zNorm(Q)$

for $i := 1$ **to** $n - m + 1$ **do**

$\hat{T}_{i,m} := zNorm(T_{i,m})$

$D(i) := \sqrt{\sum_{j=1}^{m} (\hat{t}_j - \hat{q}_j)^2}$

return D

Сложность $O(nm)$,

на каждой итерации нужно
$2m$ арифметических операций,
т.к. подпоследовательность посещается дважды:
для нормализации и для вычисления расстояния

Алгоритм грубой силы (Brute Force)

Algorithm $BruteForce$

Input: запрос $Q \in \mathbb{R}^m$, ряд T

Output: профиль расстояния $D \in \mathbb{R}^{n-m+1}$

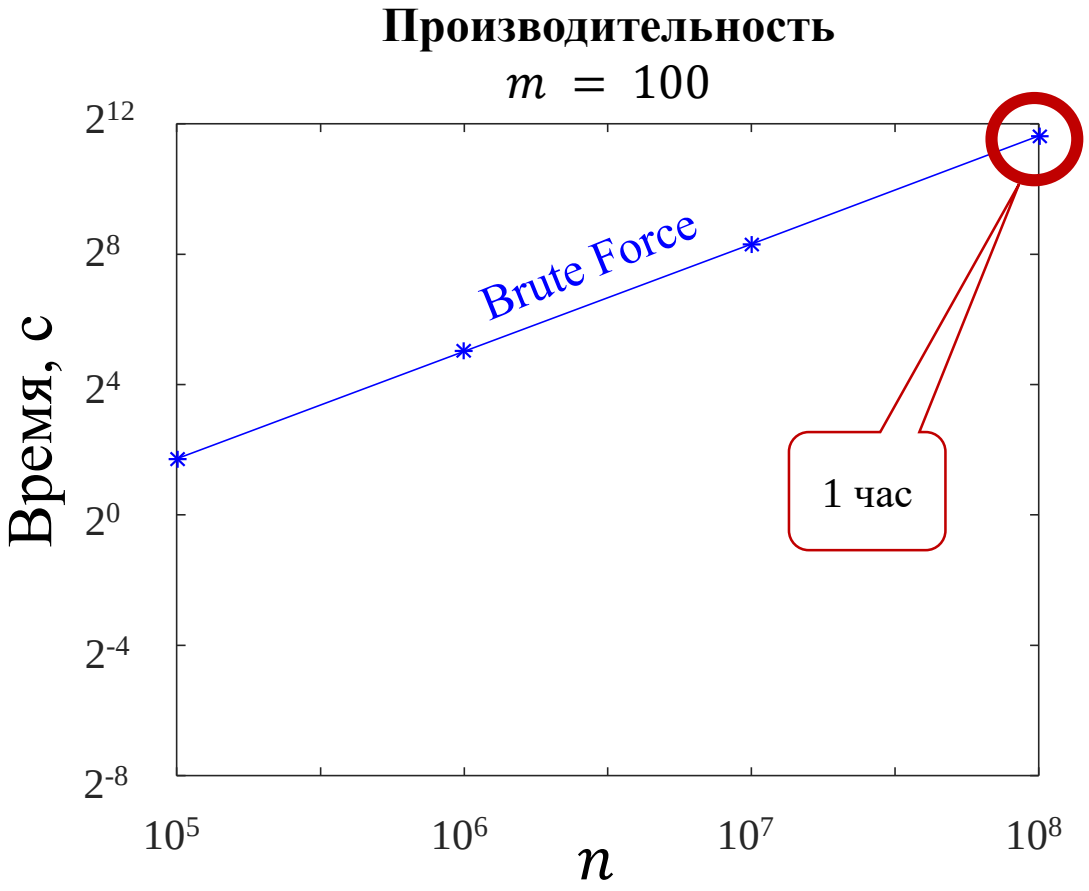
$D := \bar{0}; \hat{Q} := zNorm(Q)$

for $i := 1$ **to** $n - m + 1$ **do**

$\hat{T}_{i,m} := zNorm(T_{i,m})$

$$D(i) := \sqrt{\sum_{j=1}^{m} (\hat{t}_j - \hat{q}_j)^2}$$

return D



Улучшение алгоритма Brute Force: Нормализация «на лету»

- Можно не делать z-нормализацию на каждой итерации за счет формулы связи ED и corr
- Запрос Q предварительно z-нормализован, значит, $\boldsymbol{\mu_Q = 0}$ и $\boldsymbol{\sigma_Q = 1}$. Тогда

$$\mathrm{ED}(\hat{T}_{i,m}, \hat{Q}) = \sqrt{2m(1 - \frac{\langle T_{i,m}, Q\rangle - m\mu_{T_{i,m}}\boldsymbol{\mu_Q}}{m\sigma_{T_{i,m}}\boldsymbol{\sigma_Q}})} = \sqrt{2m(1 - \frac{\langle T_{i,m}, Q\rangle}{m\boldsymbol{\sigma_{T_{i,m}}}})}$$

- Стандартное отклонение скользящего окна, $\sigma_{T_{i,m}}$, может быть вычислено за один проход за счет применения *кумулятивных (накапливаемых) сумм*:
  - $C_i = \sum_{k=1}^{i} t_k, \quad C2_i = \sum_{k=1}^{i} t_k^2$
  - $S_i = C_{i+m} - C_i, S2_i = C2_{i+m} - C2_i$
  - $\boldsymbol{\sigma_{T_{i,m}}} = \sqrt{\frac{S2_i}{m} - \left(\frac{S_i}{m}\right)^2}$

Алгоритм JITnorm (Just In Time z-normalization)

Algorithm $JITnorm$

Input: запрос $Q \in \mathbb{R}^m$, ряд T

Output: $D \in \mathbb{R}^{n-m+1}$ (профиль расст-я)

$D := \bar{0}; \hat{Q} := zNorm(Q)$

$Sigma := movstd(T, m)$

for $i := 1$ **to** $n - m + 1$ **do**

$$D(i) := \sqrt{2 \cdot \left(m - \sum_{j=1}^{m} \frac{t_{i+j-1} \cdot \hat{q}_j}{Sigma(i)}\right)}$$

return D

Вычисление станд. откл. с однократным посещением элементов T

Сложность $O(nm)$

function $movstd$

Input: ряд T, длина подпосл-ти m

Output: массив станд. откл. $\sigma \in \mathbb{R}^{n-m+1}$

$S, S2 \in \mathbb{R}^{n+1}$

Массивы кумулят. сумм

$S := \bar{0}; \sigma := \bar{0}$

for $i := 2$ **to** $n + 1$ **do**

$S(i) := S(i-1) + t_{i-1}$

$S2 := S^2$

for $i := m + 1$ **to** $n + 1$ **do**

$j := i - m$

$$\sigma(j) := \sqrt{\frac{1}{m}\left(S2(i) - S2(j)\right) - \left(\frac{1}{m}(S(i) - S(j))\right)^2}$$

return σ

Алгоритм JITnorm (Just In Time z-normalization)

Algorithm $JITnorm$

Input: $Q \in \mathbb{R}^m$ (запрос), T (ряд)

Output: $D \in \mathbb{R}^{n-m+1}$ (профиль расст-я)

$D := \bar{0}; \hat{Q} := zNorm(Q)$

$Sigma := movstd(T, m)$

for $i := 1$ **to** $n - m + 1$ **do**

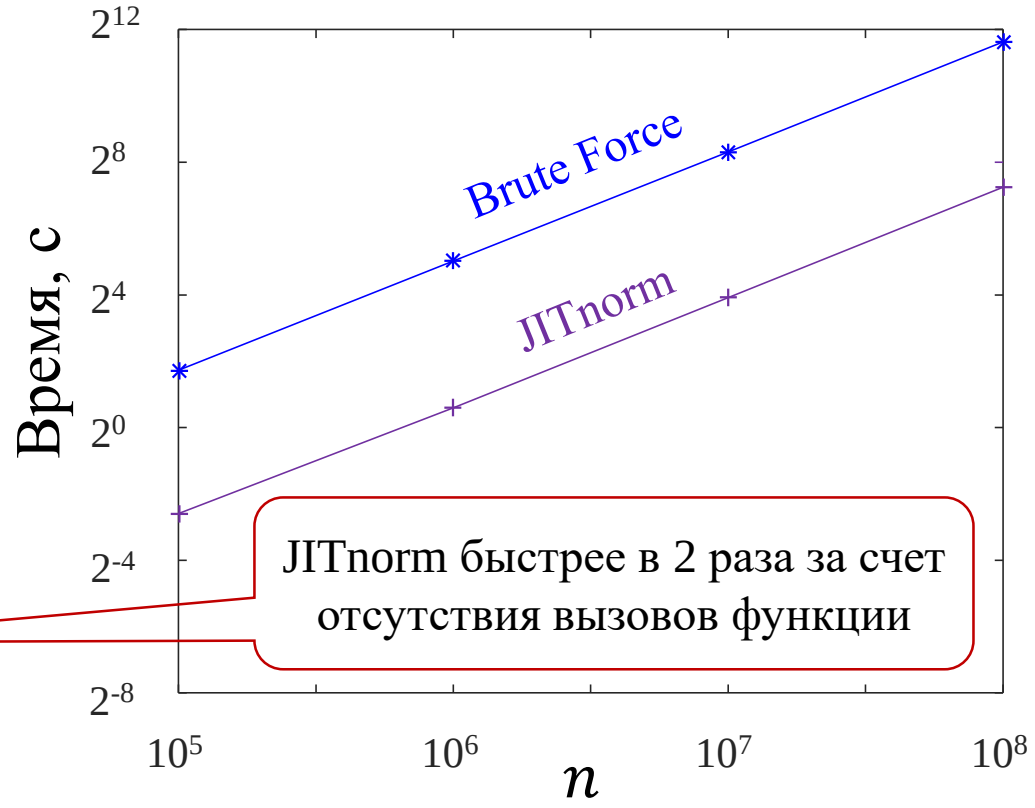
$$D(i) := \sqrt{2 \cdot \left(m - \sum_{j=1}^{m} \frac{t_{i+j-1} \cdot \hat{q}_j}{Sigma(i)}\right)}$$

return D

Сложность $O(nm)$

Производительность

$m = 100$



JITnorm быстрее в 2 раза за счет отсутствия вызовов функции

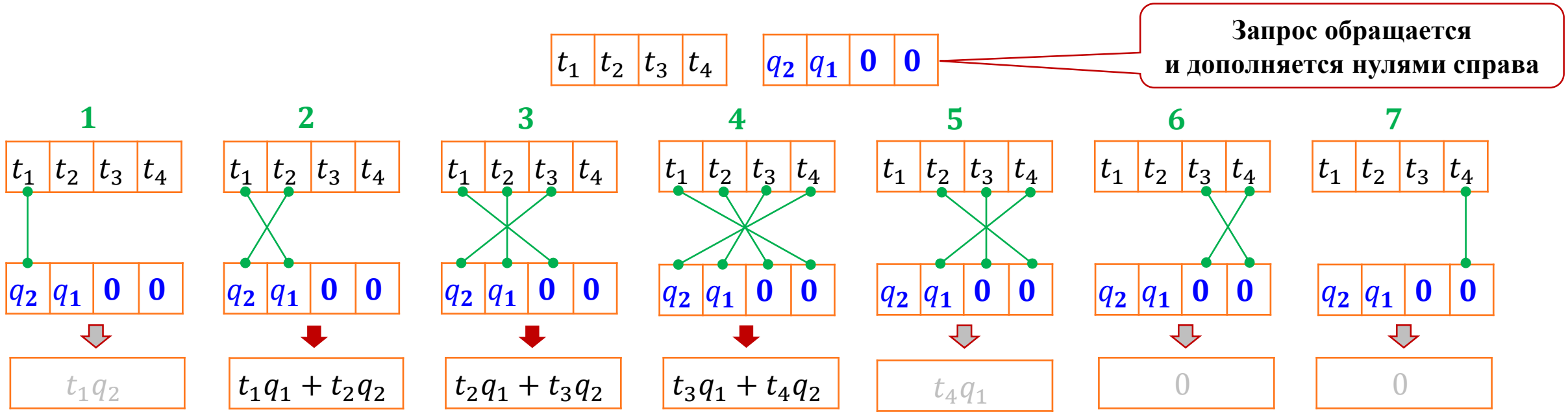
Улучшение алгоритма JITnorm: Свертка последовательностей

- Можно вычислить $\langle T_{i,m}, Q \rangle$ для всех $i \in [1, n-m+1]$ за $O(n\log_2 n)$, применяя свертку
- Если T и Q рассматривать как последовательности коэффициентов двух многочленов от одной и той же переменной, то свертка T и Q эквивалентна умножению многочленов
- $A = \{a_i\}_{i=0}^{N_A}$, $B = \{b_i\}_{i=0}^{N_B}$, ***свертка*** $\boldsymbol{A * B = C}$, $\boldsymbol{C = \{c_i\}_{i=0}^{N_A+N_B-1}}$, где $\boldsymbol{c_i = \sum_{j=0}^{i} a_j b_{i-j}}$, причем $a_j = 0$ при $j \geq N_A$ и $b_{i-j} = 0$ при $i - j \geq N_B$
- Пример: $T = (t_1, t_2, t_3, t_4)$ и $Q = (q_1, q_2)$
  - $|T * Q| = 4 + 2 - 1 = 5$
  - $(T * Q)_0 = t_1 q_1$
  - $(T * Q)_1 = t_1 q_2 + t_2 q_1$
  - $(T * Q)_2 = t_2 q_2 + t_3 q_1$
  - $(T * Q)_3 = t_3 q_2 + t_4 q_1$
  - $(T * Q)_4 = t_4 q_2$

	0	1	2	3	4
$\boldsymbol{c_0}$	$a_0 b_0$				
$\boldsymbol{c_1}$	$a_0 b_1$	$a_1 b_0$			
$\boldsymbol{c_2}$	$a_0 b_2$	$a_1 b_1$	$a_2 b_0$		
$\boldsymbol{c_3}$	$a_0 b_3$	$a_1 b_2$	$a_2 b_1$	$a_3 b_0$	
$\boldsymbol{c_4}$	$a_0 b_4$	$a_1 b_3$	$a_2 b_2$	$a_3 b_1$	$a_4 b_0$

Улучшение алгоритма JITnorm: Применение свертки

- Для вычисления скалярных произведений выполняется свертка ряда и реверс-версии запроса, которая дополнена нулями справа
- Пример: $T = (t_1, t_2, t_3, t_4)$ и $Q = (q_1, q_2)$,
  $\langle T_{1,2}, Q \rangle = t_1 q_1 + t_2 q_2, \quad \langle T_{2,2}, Q \rangle = t_2 q_1 + t_3 q_2, \quad \langle T_{3,2}, Q \rangle = t_3 q_1 + t_4 q_2$



Алгоритм MASS1

Algorithm $MASS1$

Input: запрос $Q \in \mathbb{R}^{2m}$, ряд T

Output: профиль $D \in \mathbb{R}^{n-m+1}$

$QT \in \mathbb{R}^{n-m+1}$

$D := \bar{0}; \hat{Q}_{1,m} := zNorm(Q_{1,m});$

$\hat{Q}_{1,m} := Reverse(\hat{Q}_{1,m}); \hat{Q}_{m+1,m} := \bar{0}$

$Sigma := movstd(T, m)$

$QT := \text{conv}(T, \hat{Q})$

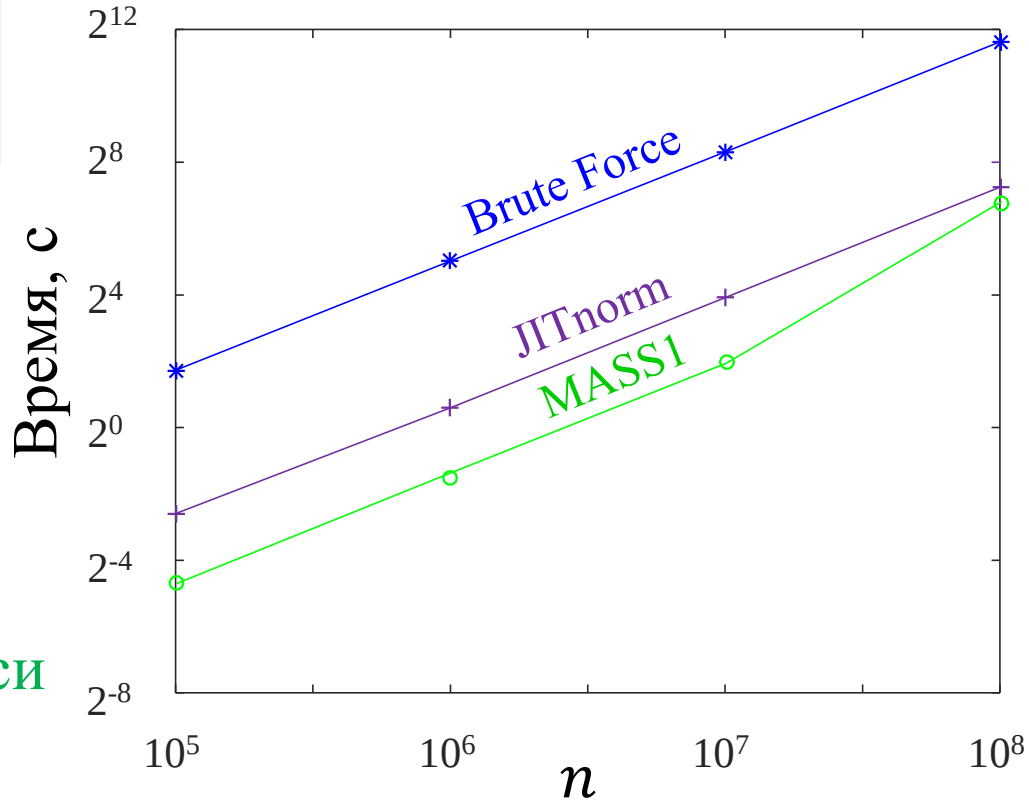
$D := \sqrt{2 \cdot (m - \frac{QT(m..n)}{Sigma})}$ // Векторная форма записи

return D

Сложность $O(n\log_2 n)$ **не зависит от** m

Производительность

$m = 100$



Улучшение алгоритма MASS1: Дискретное преобразование Фурье

- Свертка двух последовательностей с помощью двух ДПФ аргументов свертки и одного обратного ДПФ (ОДПФ) результата свертки
- Вычисление свертки для $A = \{a_i\}_{i=0}^{N_A}$, $B = \{b_i\}_{i=0}^{N_B}$,
  1. Дополнение A и B справа нулями до длины $N_C = \mathbf{2} \cdot \max(N_A, N_B)$: $Adp := DoublePad(A)$; $Bdp := DoublePad(B)$
  2. Выполнение ДПФ для A и B: $FA := DFT(Adp)$; $FB := DFT(Bdp)$
  3. Вычисление поэлементного произведения результатов ДПФ: $FC := FA \cdot FB$
  4. Выполнение ОДПФ для FC: $C := IDFT(FC)$
  5. Результат свертки: $\{c_i\}_{i=0}^{N_C/2}$

Жан-Батист Жозеф Фурье (Jean-Baptiste Joseph Fourier)
1768-1830

Улучшение алгоритма MASS1: Дискретное преобразование Фурье

- $X = \{x_k\}_{k=0}^{N_X}, Xdp = \{x_k\}_{k=0}^{N}, N = 2N_X$
- ДПФ: $FX = DFT(Xdp)$

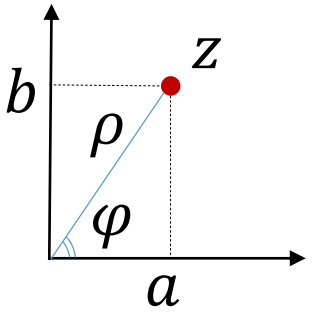
$$fx_k = \sum_{\ell=0}^{N-1} x_\ell \cdot e^{-\boldsymbol{i}\frac{2\pi k\ell}{N}}$$

- ОДПФ: $X = IDFT(Xdp)$

$$x_k = \frac{1}{N}\sum_{\ell=0}^{N-1} fx_\ell \cdot e^{\boldsymbol{i}\frac{2\pi k\ell}{N}}$$

Формы записи комплексного числа

$z = a + \boldsymbol{i}b, \boldsymbol{i}^2 = -1$



$$z = \rho(\cos\varphi + \boldsymbol{i}\sin\varphi),$$
$$\rho = |z| = \sqrt{a^2 + b^2},$$
$$\cos\varphi = \frac{a}{\rho}, \sin\varphi = \frac{b}{\rho}$$

$$e^z = e^a(\cos b + \sin b)$$

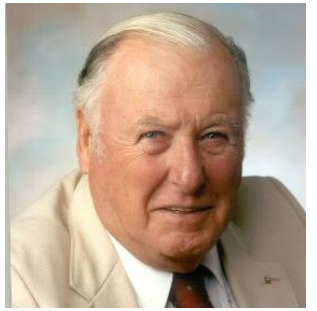
Улучшение алгоритма MASS1: Быстрое преобразование Фурье

- ДПФ и ОДПФ выполняются за $O(n\log_2 n)$ с помощью БПФ (Fast Fourier Transform)
- Открыто К. Гауссом в 1805 г. для интерполяции траекторий астероидов, опубликовано после его смерти
- Алгоритм Гуда—Томаса (1958, 1963) для БПФ в случае $n = n_1 \cdot n_2$, где n_1 и n_2 взаимно просты
- Алгоритм Кули—Тьюки (1965) для БПФ в случае произвольного n; описана реализация на ЭВМ

Карл Гаусс
(Johann Carl Friedrich Gauß)
1777-1855

Ирвинг Гуд
(Irving John Good)
1916-2009

Люэлин Томас
(Llewellyn Hilleth Thomas)
1903-1992



Джеймс Кули
(James William Cooley)
1926-2016

Джон Тьюки
(John Wilder Tukey)
1915-2000

Алгоритм MASS2

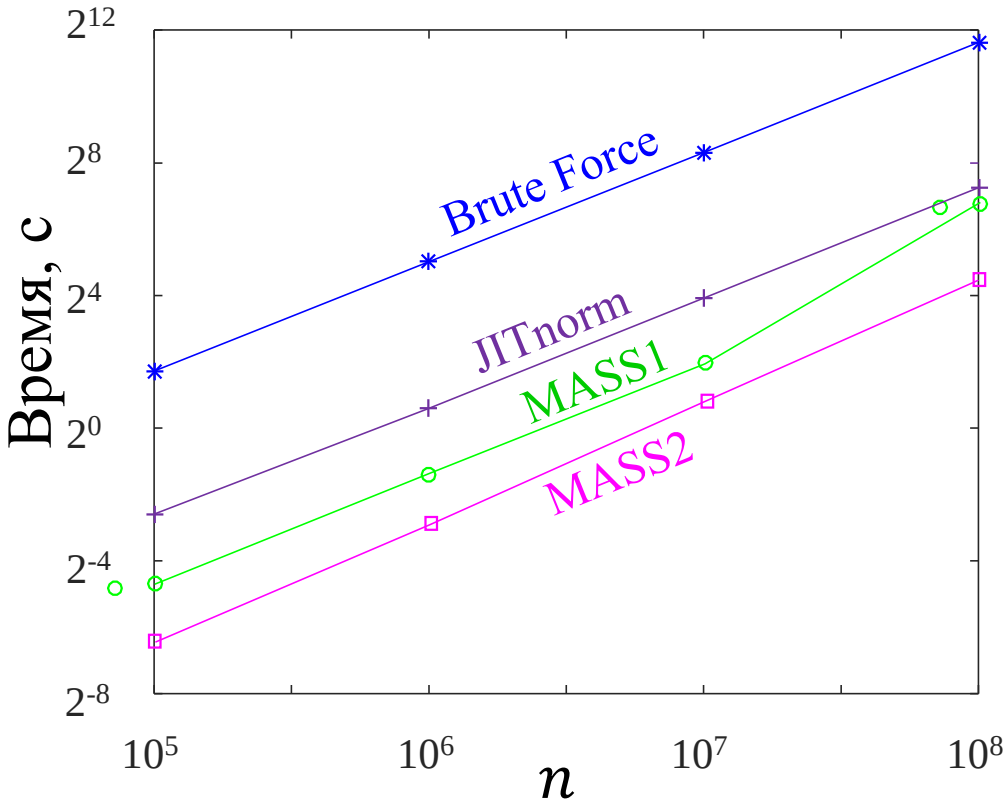
Algorithm $MASS2$
Input: запрос $Q \in \mathbb{R}^{2m}$, ряд T
Output: профиль $D \in \mathbb{R}^{n-m+1}$
$QT \in \mathbb{R}^{n-m+1}$

$D := \bar{0}; \hat{Q}_{1,m} := zNorm(Q_{1,m})$
$\hat{Q}_{1,m} := Reverse(\hat{Q}_{1,m})$
$\hat{Q}_{m+1,m} := \bar{0}$
$Sigma := movstd(T, m)$
$QT := IFFT(FFT(T) \cdot FFT(\hat{Q}))$
$D := \sqrt{2 \cdot (m - \frac{QT(m..n)}{Sigma})}$
return D

Сложность $O(n\log_2 n)$ **не зависит от** $\boldsymbol{m}$

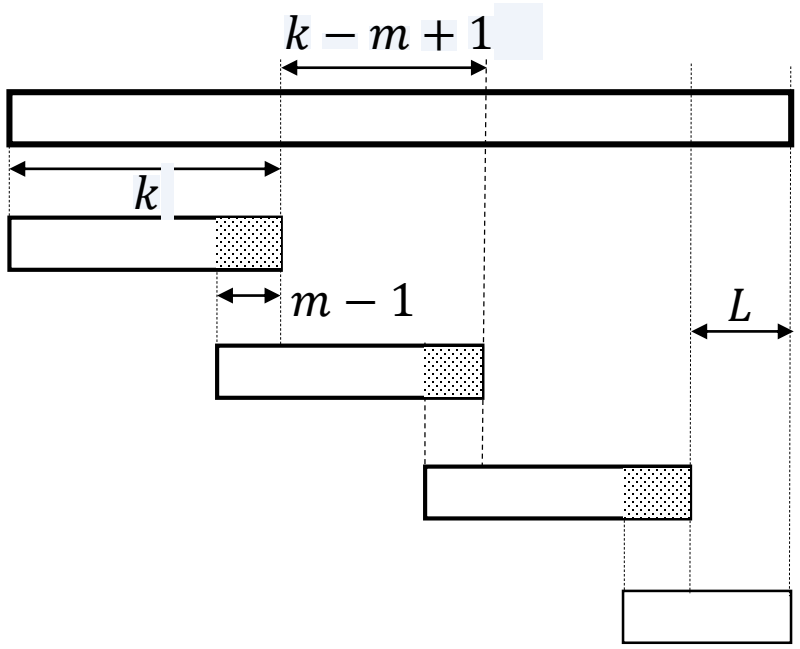
К T и Q *не* применяется $DoublePad$, т.к. **вычисляется лишь половина свертки**

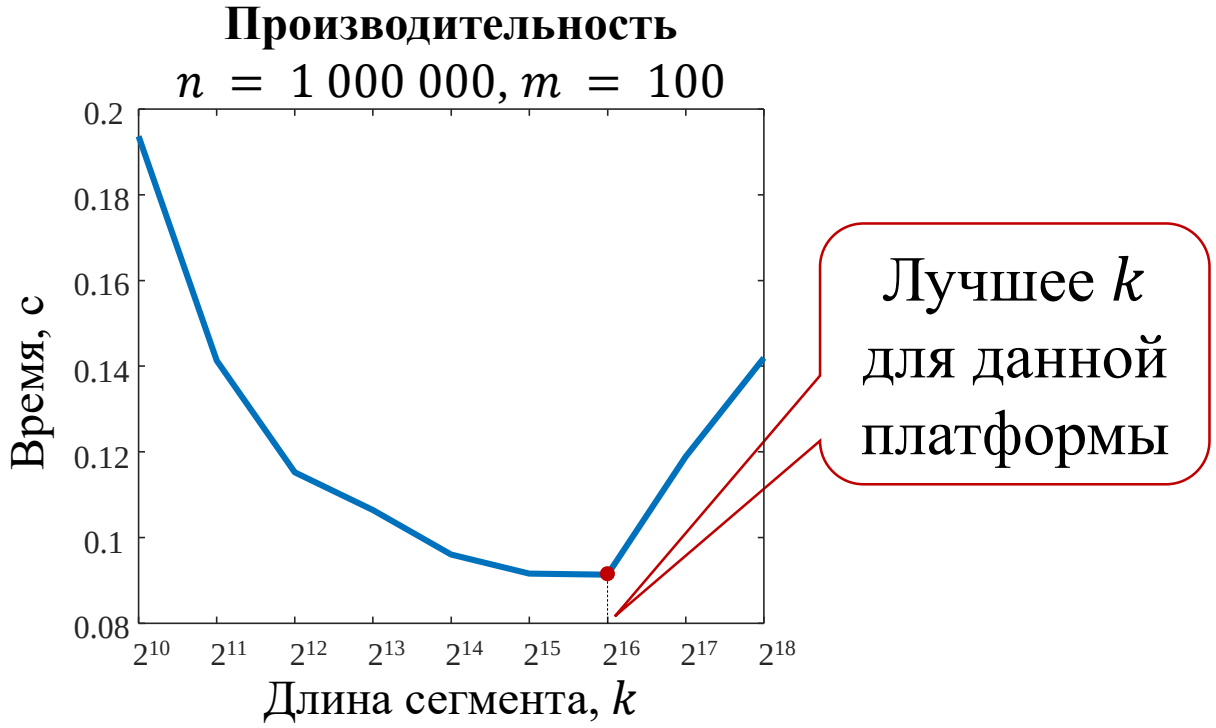
Производительность
$m = 100$



Улучшение алгоритма MASS2: Сегментация ряда

- Вместо обработки ряда T целиком (он может не помещаться в памяти) будем делать это посегментно
- Длина каждого сегмента (кроме, возможно, последнего) – степень двойки (параметр, который зависит от аппаратной платформы)
- Сегменты перекрываются на $m - 1$ элементов



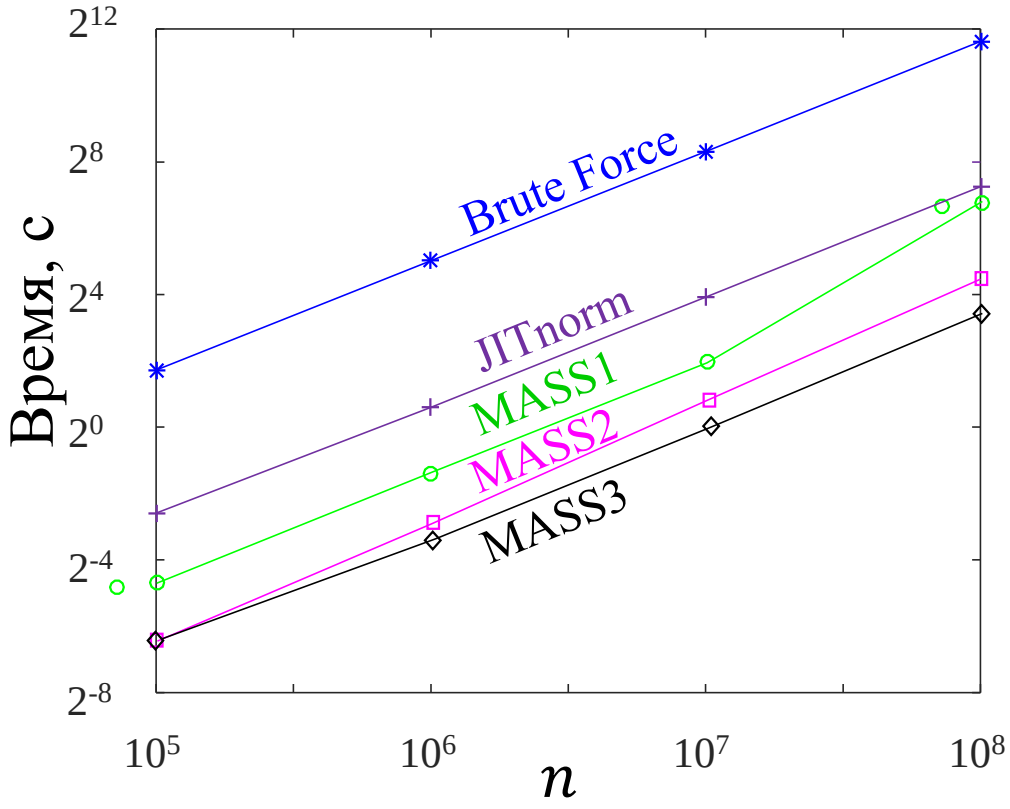


Алгоритм MASS3

Algorithm *MASS3*
Input: запрос $Q \in \mathbb{R}^{m+k}$, ряд T, длина сегмента k
Output: профиль $D \in \mathbb{R}^{n-m+1}$
$QT \in \mathbb{R}^{n-m+1}$, $T' \in \mathbb{R}^{k}$

$D := \bar{0}$; $\hat{Q}(1..m) := zNorm(Q(1..m))$;
$\hat{Q}(1..m) := Reverse(\hat{Q}(1..m))$
$\hat{Q}(m+1..k) := \bar{0}$; $T' := \bar{0}$
for $i := 1$ **to** $n - m + 1$ **step** $k - m + 1$ **do**
  $j := 1 + k - 1$; $T'(1..k) := T(i..j)$
  if $j > n$ **then** // Обработка последнего сегмента
    $j := n$; $T' := \bar{0}$; $T'(1..j - i + 1) := T(i..j)$
  $Sigma := movstd(T', m)$
  $QT := IFFT(FFT(T') \cdot FFT(\hat{Q}))$
  $D(i..j - m + 1) := \sqrt{2 \cdot (m - \frac{QT}{Sigma})}$
return D

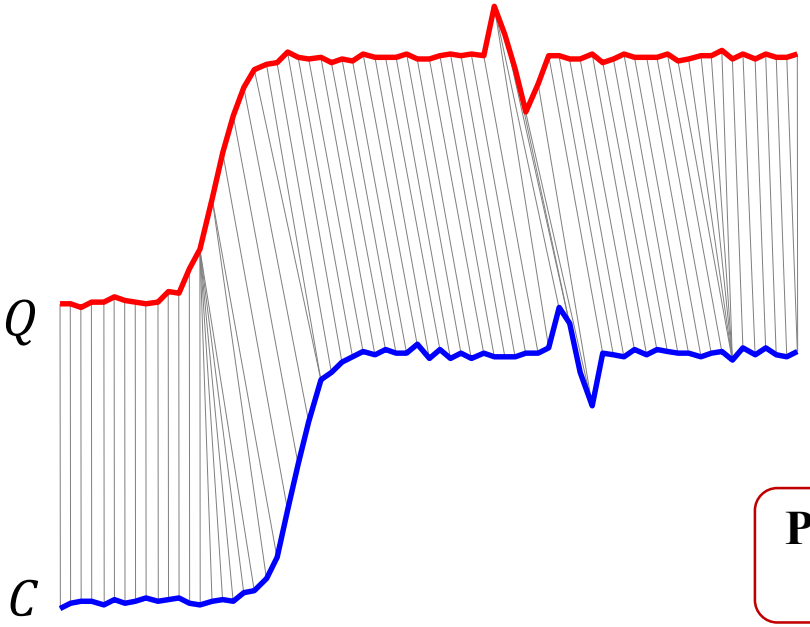
Производительность
$m = 100$

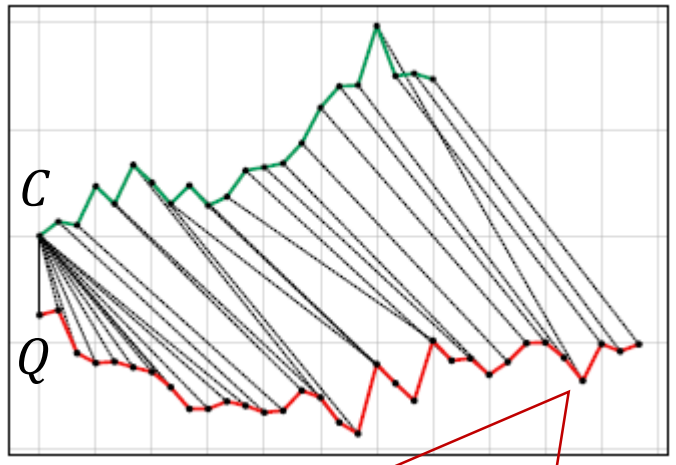


Содержание

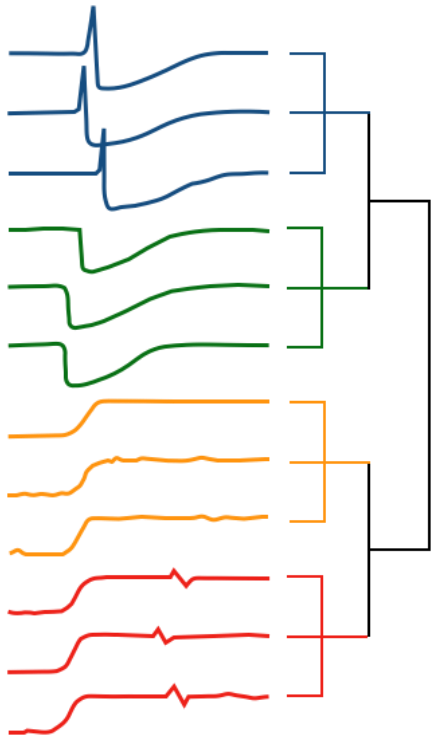
Динамическая трансформация времени: DTW, Dynamic Time Warping*

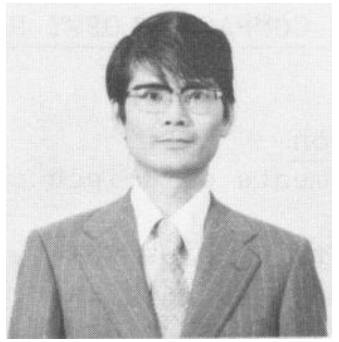
- **Не метрика**: не выполняется неравенство треугольника!
- Вычислительная сложность $O(m^2)$
- Сравнение рядов по принципу «один к много», адекватно учитывает форму рядов





Ряды могут быть разной длины!
Выч. сложность $O(m_1 \cdot m_2)$



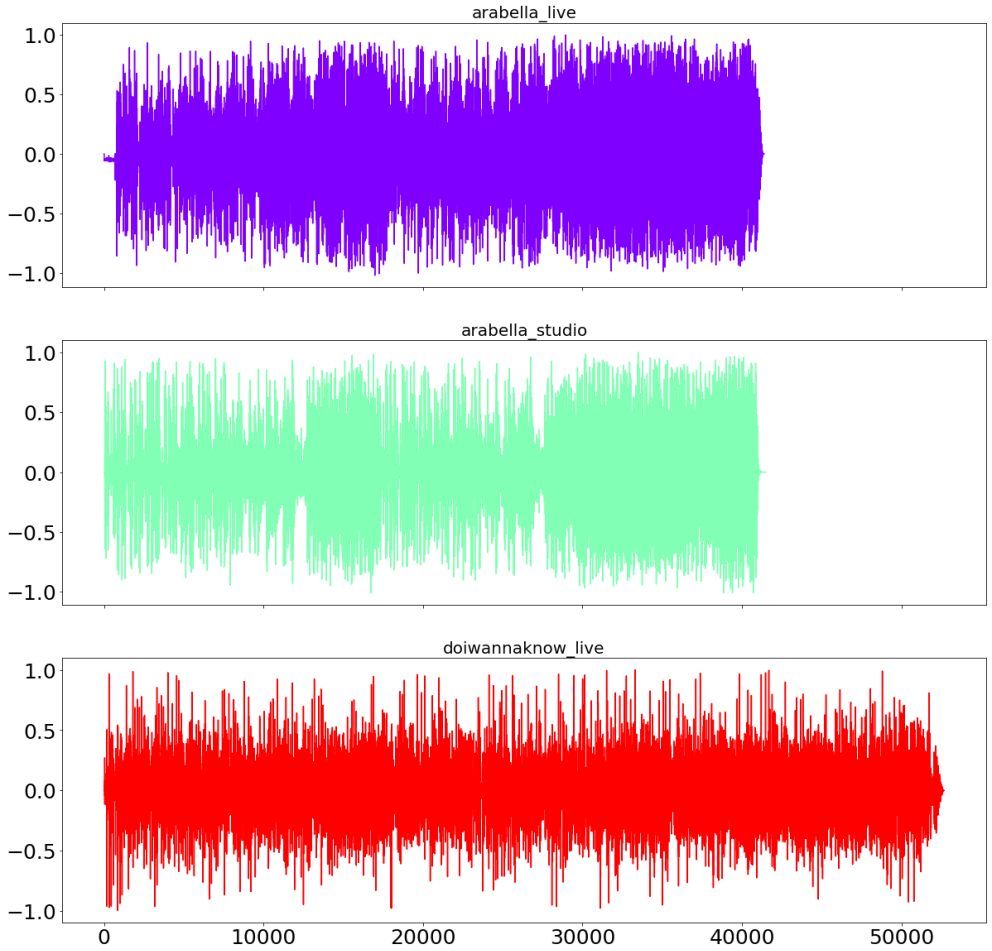


Хироаки Сако
(Hiroaki Sakoe)

Сэйби Чиба
(Sabi Chiba)

* Sakoe H., Chiba S. Dynamic programming algorithm optimization for spoken word recognition. IEEE Trans. on Acoustics, Speech, and Signal Processing, 1978. 26(1), 43-49. DOI: 10.1109/TASSP.1978.1163055

Применение DTW: распознавание речи



Arabella (studio)

Arabella (live)

Do I Wanna Know (live)

ED (норм.)	Arabella (live)	Do I Wanna Know (live)
Arabella (studio)	0.043	0.038

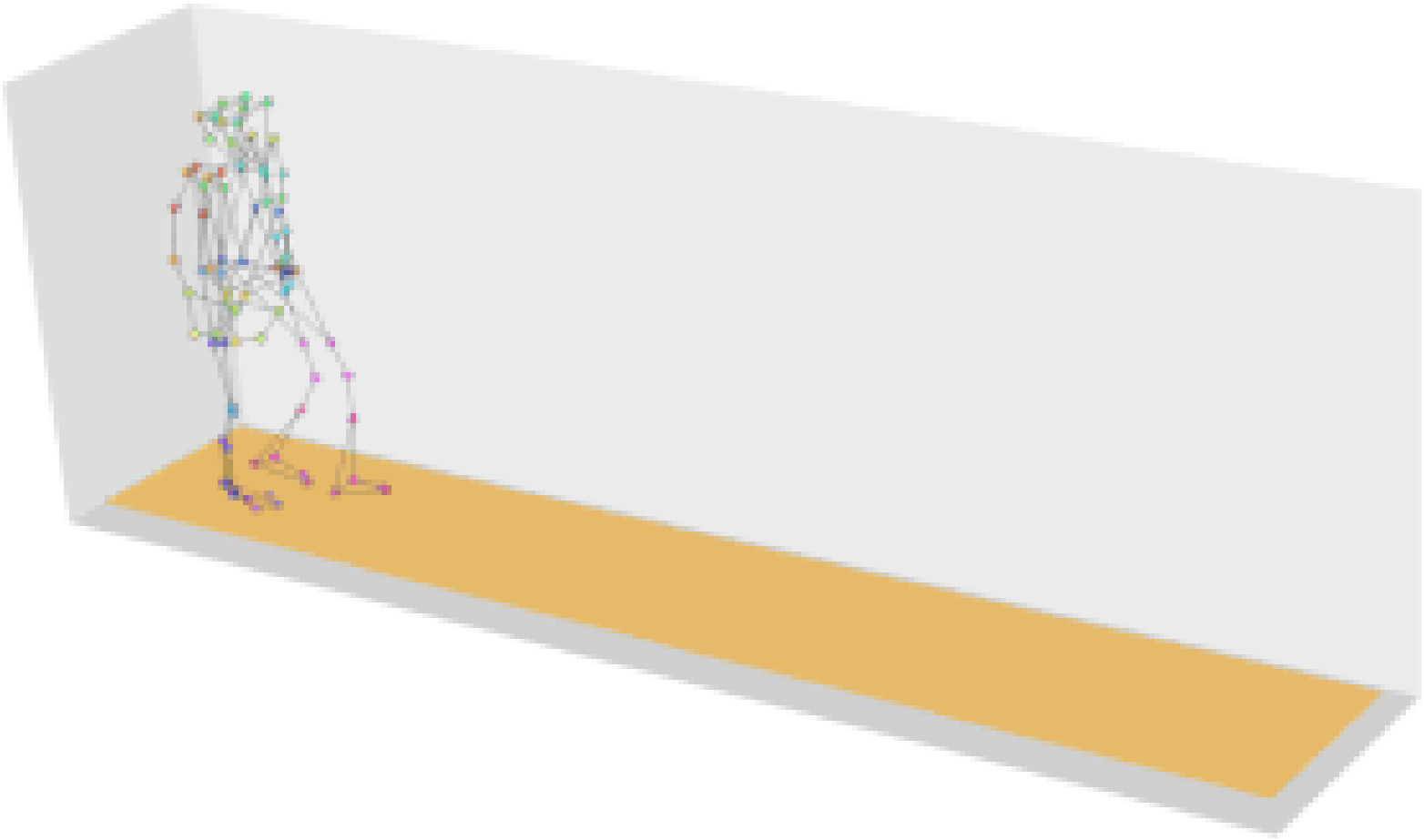
ED не отличает
разные записи разных песен!

DTW отличает
разные записи одной песни!

DTW (норм.)	Arabella (live)	Do I Wanna Know (live)
Arabella (studio)	0.82	1

Mora P. Dynamic Time Warping: Explanation and extensive testing on audio and tabular data. URL

Применение DTW: биометрия

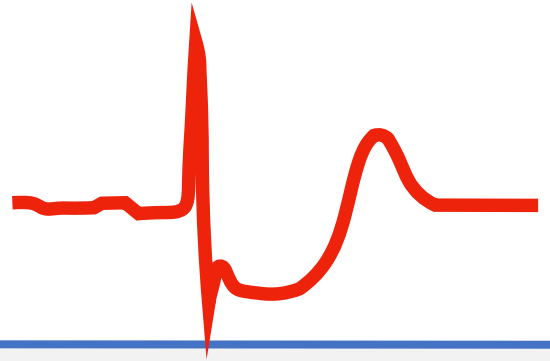


- Две записи походки одного человека с помощью системы захвата движения
- Скорость в попытках разная, но DTW помогает понять, что траектории конечностей имеют большое сходство

Применение DTW: медицина

Преждевременное сокращение желудочков

Инфаркт



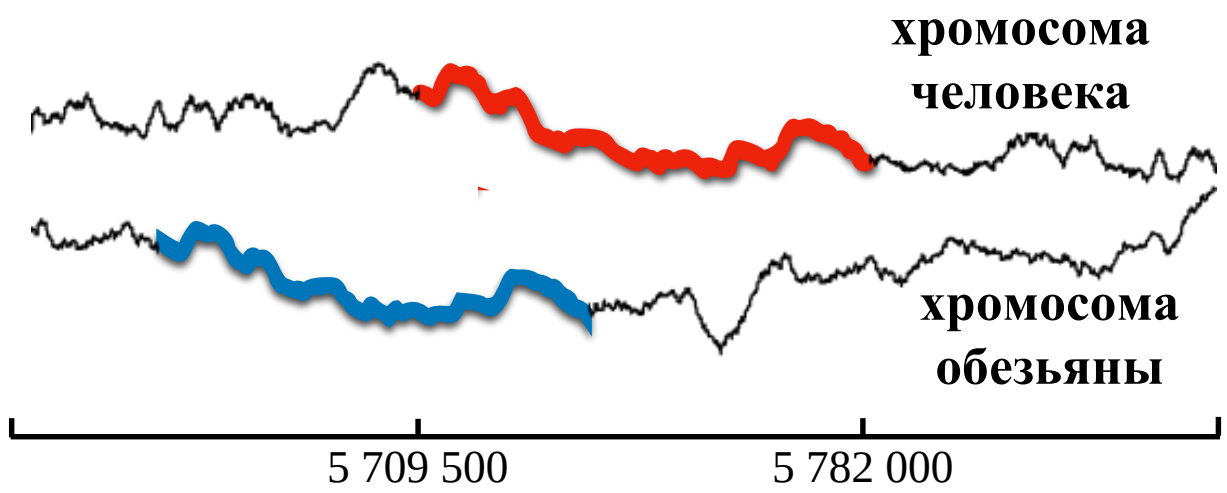
Гиперкалиемия

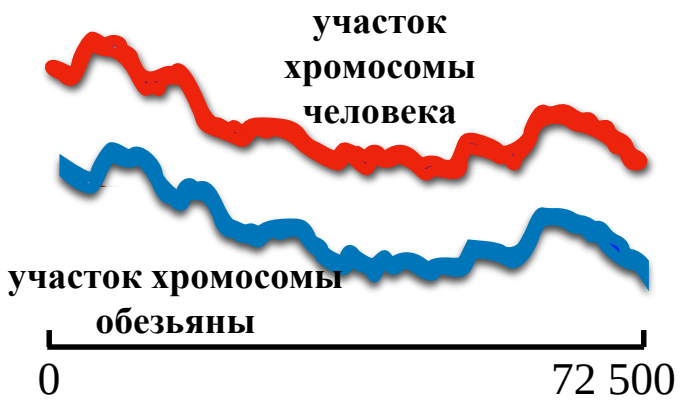
Применение DTW: генетика*

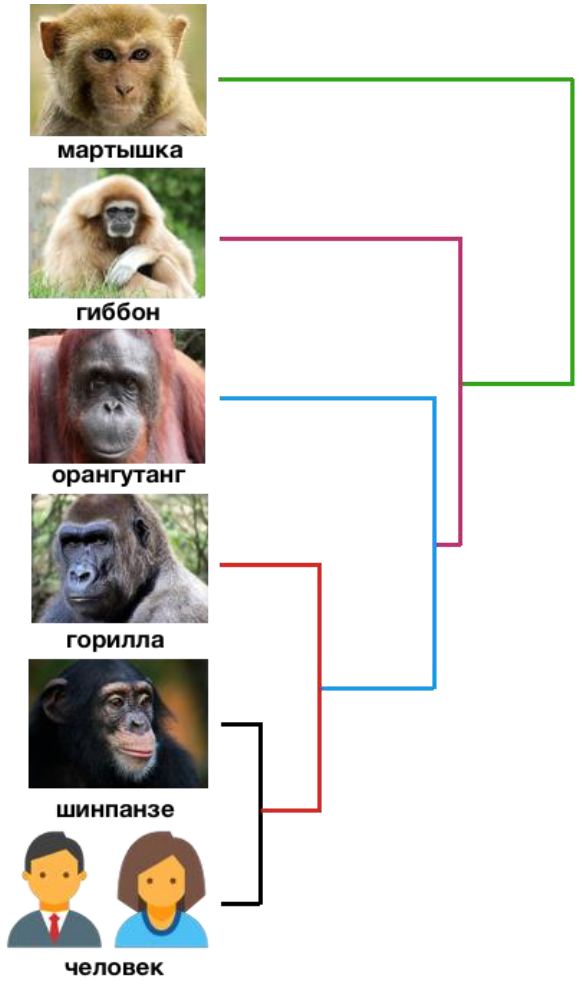
Хромосома человека:
GTCAAT…AAGAGATTTG
Хромосома обезьяны:
GGCAAT...ACAGATTTGA

Трансформация цепочки ДНК во временной ряд

$t_1 := 0$
for $i \in 1..|DNAstr|$ **do**
  case $DNAstr$ **of**
  A : $t_{i+1} := t_{i+2}$
  G : $t_{i+1} := t_{i+1}$
  C : $t_{i+1} := t_{i-1}$
  T : $t_{i+1} := t_{i-2}$
  end

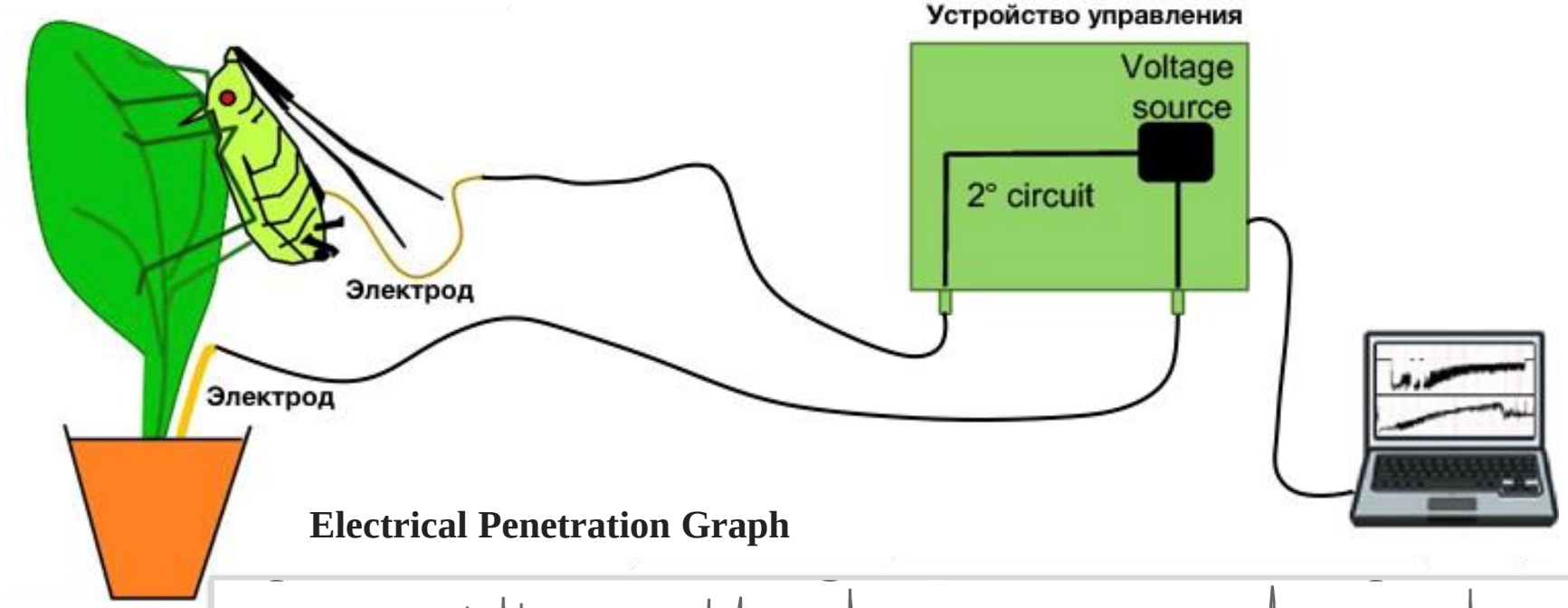




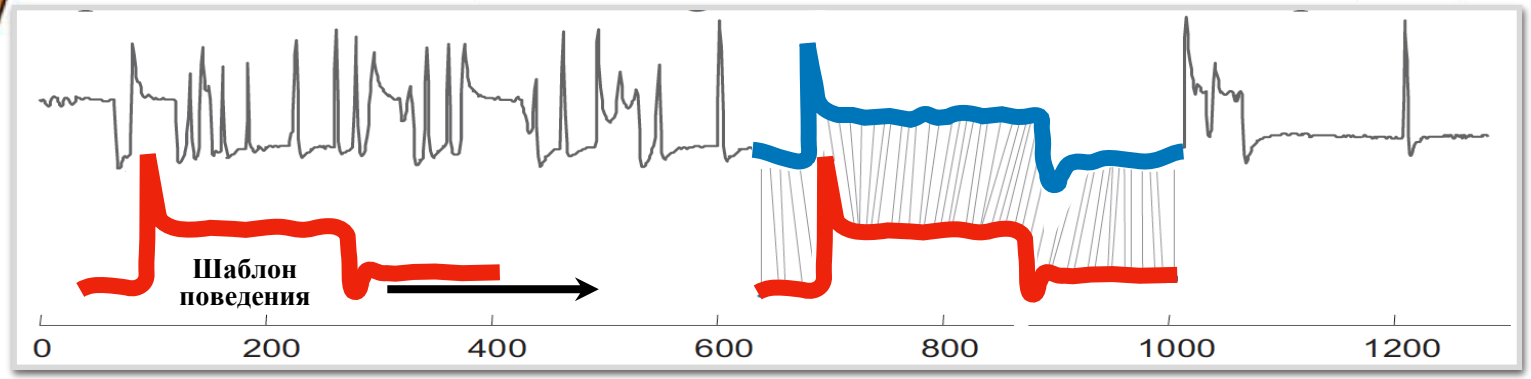


* Rakthanmanon T. *et al.* Addressing big data time series: Mining trillions of time series subsequences under Dynamic Time Warping. ACM TKDD. 2013. 7(3). 10. https://doi.org/10.1145/2500489

Применение DTW: биология

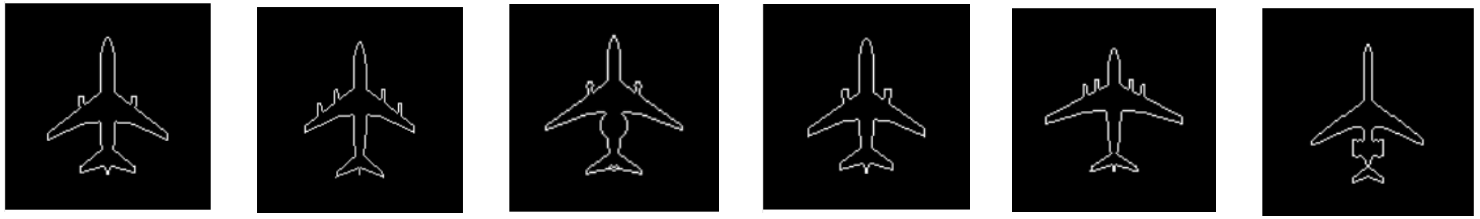


Electrical Penetration Graph

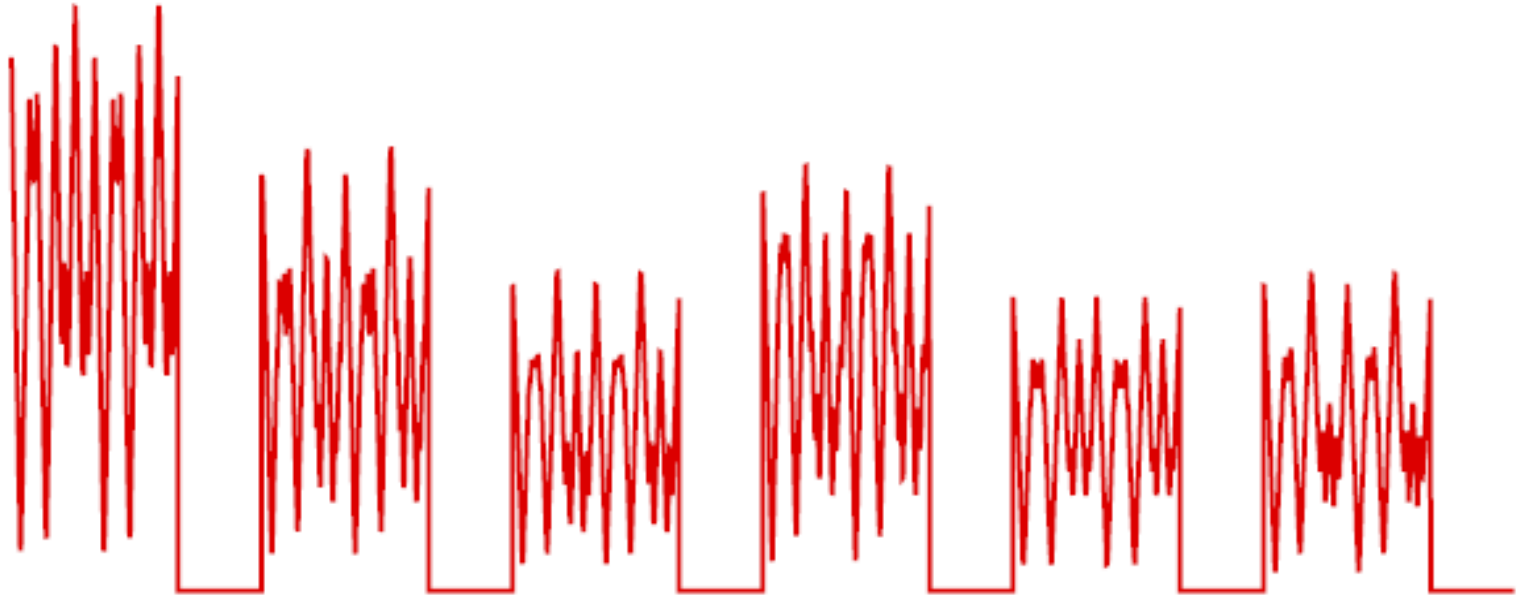


Применение DTW: военное дело

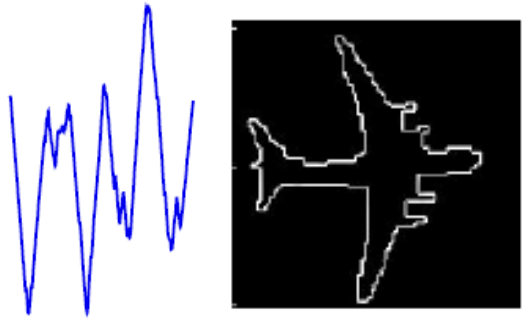
Словарь контуров



Аэрофотосъемка



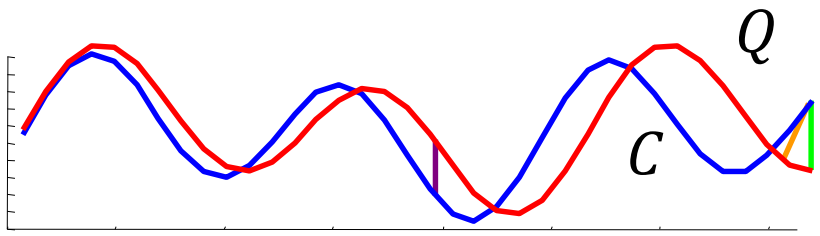
Образец поиска

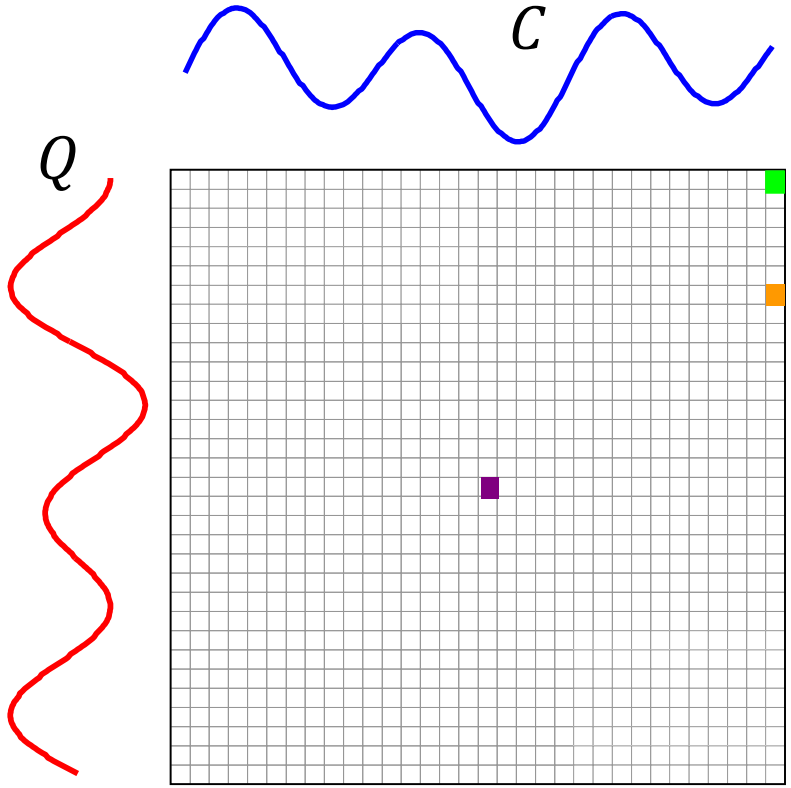


Вычисление DTW(Q, C): 1. Матрица расстояний

Построим матрицу расстояний $d \in \mathbb{R}^{m \times m}$ между точками Q и C:

$\boldsymbol{d}(i,j) = \text{Dist}(q_i, c_j)$, допустимо $\textbf{Dist}(\cdot,\cdot) = \left(\boldsymbol{q_i} - \boldsymbol{c_j}\right)^{\mathbf{2}}$ или $\text{Dist}(\cdot,\cdot) = |q_i - c_j|$



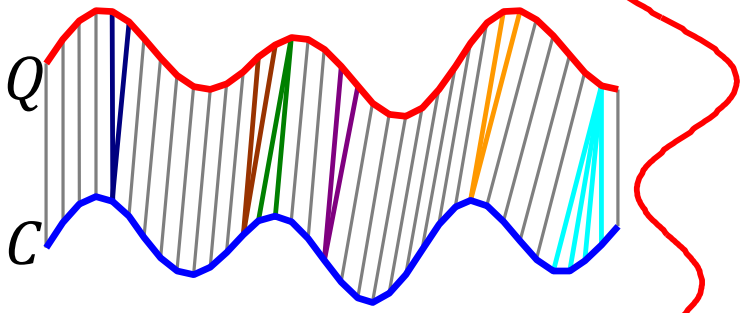


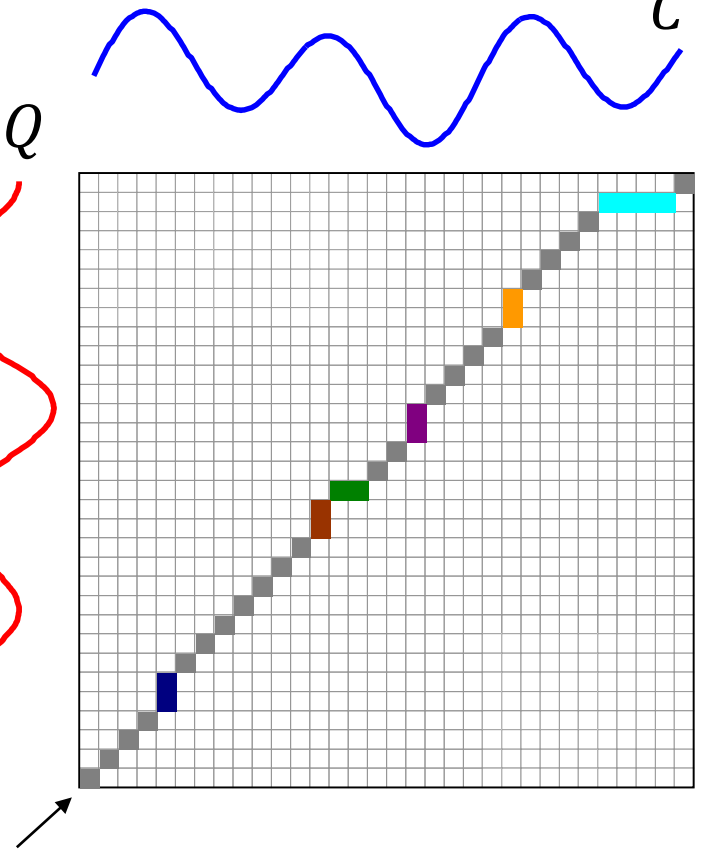
Вычисление DTW(Q, C): 2. Матрица и путь трансформации

Построим матрицу трансформации D и найдем в ней путь трансформации W, который устанавливает соответствие между Q и C, минимизируя общее расстояние между ними:

- путь $W = w_1, \dots, w_K$, длина пути $m \le K < 2m$
- элемент пути $w_k = (i,j)_k$, $d(w_k) = \text{Dist}(q_i, c_j) = d(i,j)$

1. **Полнота пути** (содержит все точки Q и C)
   $w_1 = (1,1)$, $w_K = (m,m)$
2. **Непрерывность пути** (один шаг за один раз)
   $\forall\, w_k = (i,j)$ и $w_{k+1} = (p,q)$:
   $i - p \le 1$ и $j - q \le 1$
3. **Монотонность пути** (без возврата к пройденному)
   $\forall\, w_k = (i,j)$ и $w_{k+1} = (p,q)$:
   $i - p \ge 0$ и $j - q \ge 0$





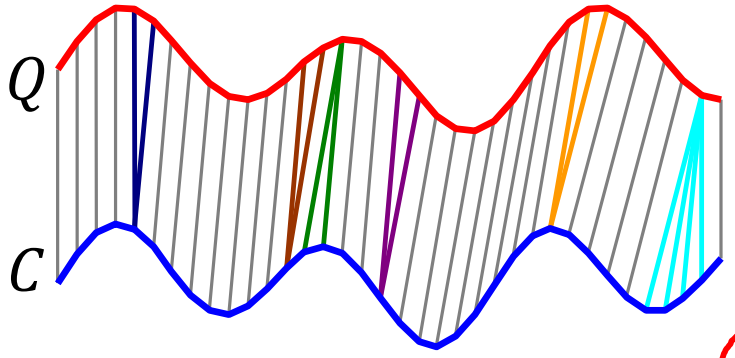
Путь трансформации *W*

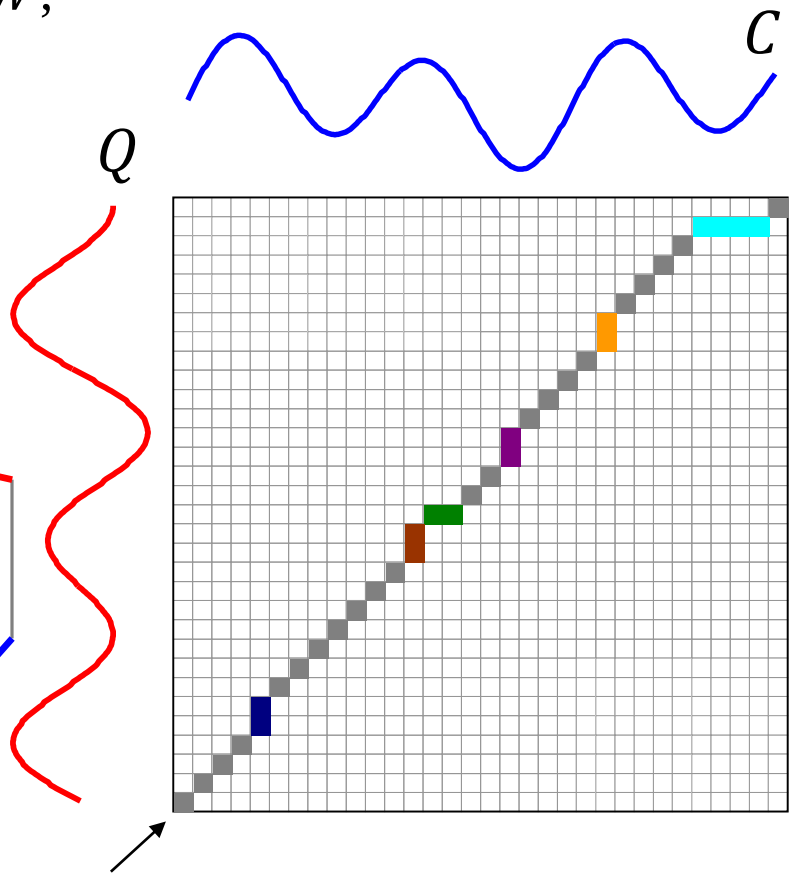
Вычисление DTW(Q, C): 2. Матрица и путь трансформации

Построим матрицу трансформации D и найдем в ней путь трансформации W, который устанавливает соответствие между Q и C, минимизируя общее расстояние между ними:

- путь $W = w_1, \dots, w_K$, длина пути $m \le K < 2m$
- элемент пути $w_k = (i,j)_k$, $d(w_k) = \text{Dist}(q_i, c_j) = d(i,j)$

$$\text{DTW}(Q,C) = \min\{\sqrt{\sum_{k=1}^{K} d(w_k)}\}$$





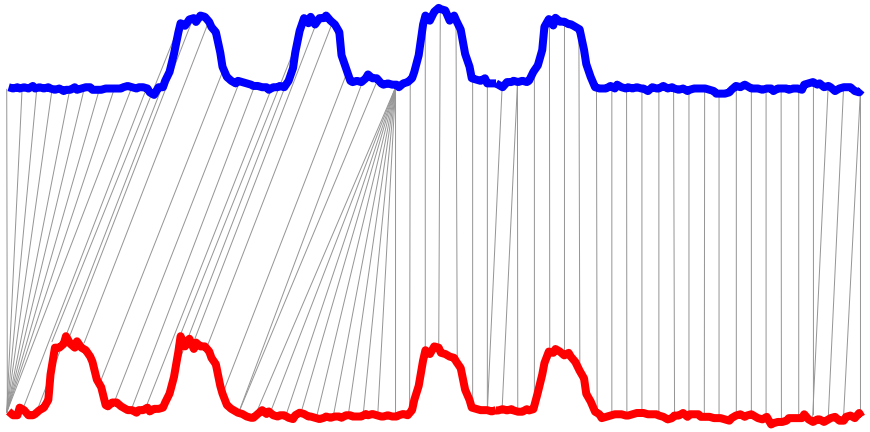
Путь трансформации *W*

NB! Далее для упрощения изложения будет рассматриваться **квадрат DTW**:
$\text{DTW}(Q,C) = \min\{\sum_{k=1}^{K} d(w_k)\}$

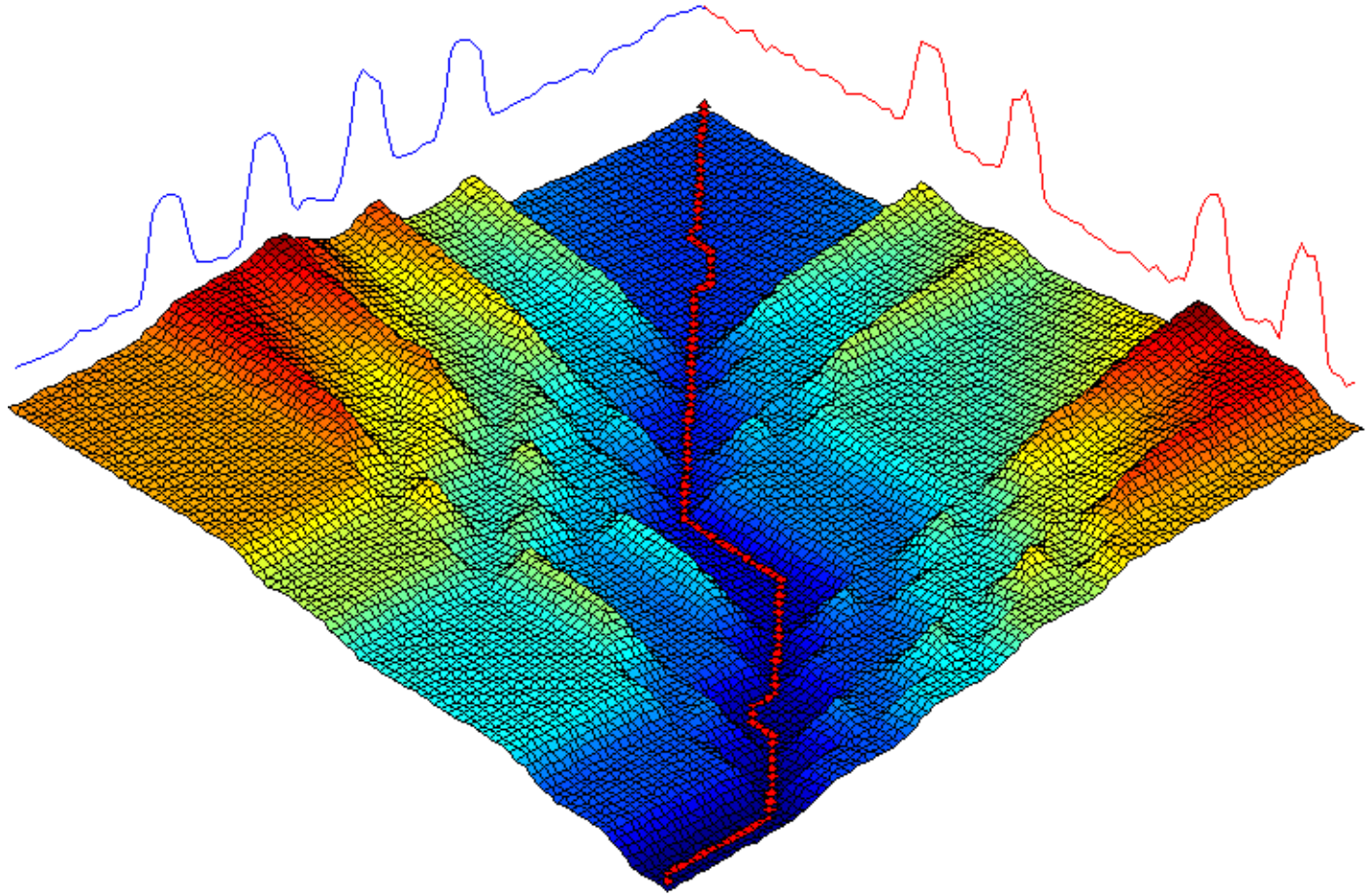
Пример матрицы и пути трансформации

Недельное энергопотребление вычислительного центра (Голландия, 1997)*

C: 4-дневная рабочая неделя,
Понедельник – выходной



Q: 4-дневная рабочая неделя,
Среда – выходной

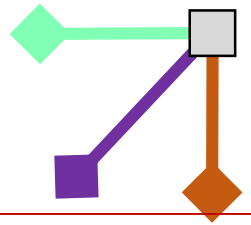


* van Wijk J.J., van Selow R.R. Cluster and calendar based visualization of time series data. INFOVIS 1999: 4-9. DOI: 10.1109/INFVIS.1999.801851

Вычисление DTW(Q, C): матрица и путь трансформации

Матрица трансформации

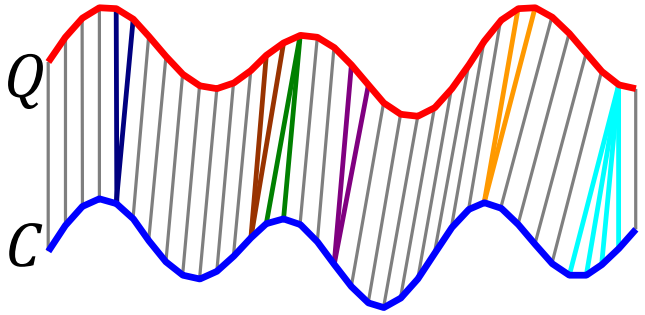
$D \in \mathbb{R}^{(m+1)\times(m+1)}$

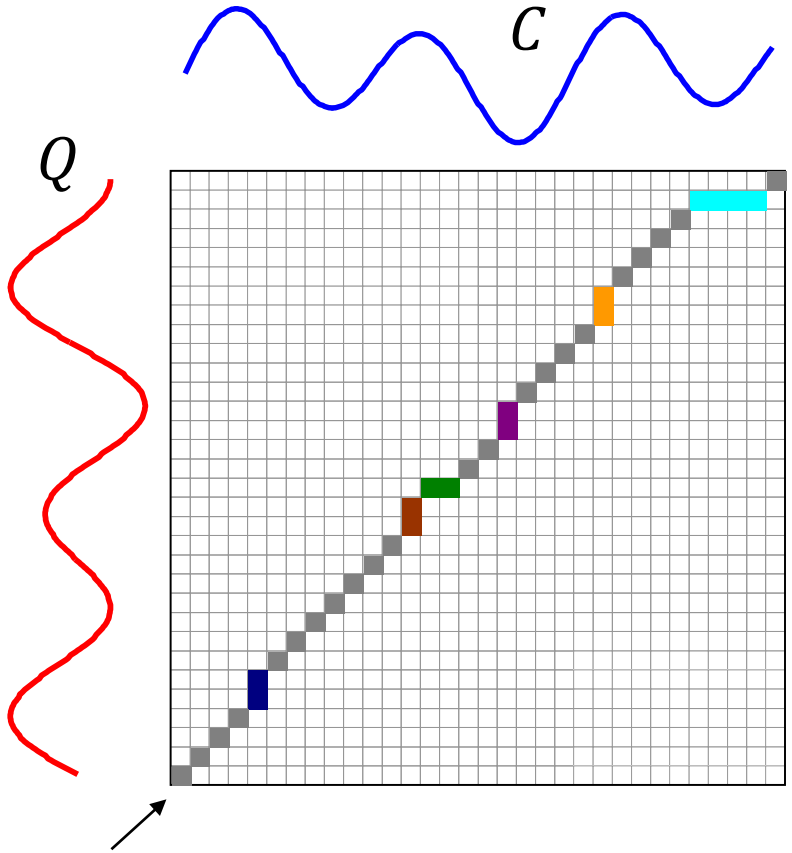


$$\mathrm{DTW}(Q,C) = D(m,m)$$

$$D(i,j) = d(i,j) + \min\{D(i-1,j), D(i-1,j-1), D(i,j-1)\}$$

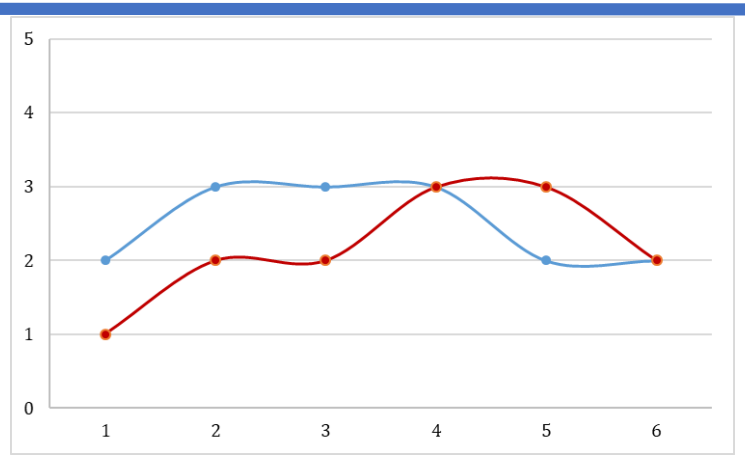
$$D(0,0) = 0, D(i,0) = D(0,j) = +\infty$$





Путь трансформации w

Пример вычисления DTW



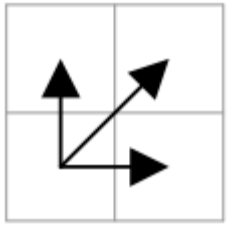
$$DTW(Q,C) = D(m,m)$$

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

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

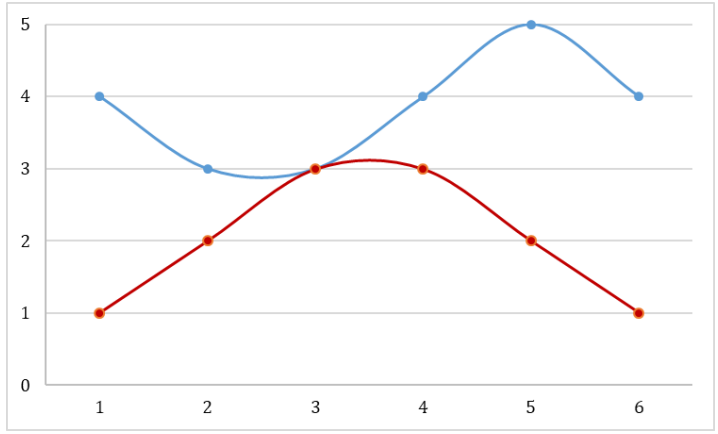
$$C = (2,3,3,3,2,2), Q = (1,2,2,3,3,2)$$

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



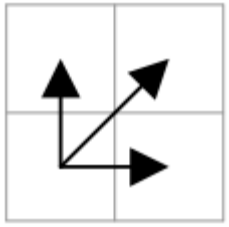
$$\mathrm{DTW}(Q, C) = D(m, m)$$

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

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

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

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
→ **j**	1	2	3	4	5	6	↑ **i**



D	4	3	3	4	5	4	
1	+∞	28	15	15	20	30	**28**	6
2	+∞	19	11	11	14	20	19	5
3	+∞	15	10	10	11	15	16	4
3	+∞	14	10	10	11	15	16	3
2	+∞	13	10	11	15	24	28	2
1	+∞	9	13	17	26	42	51	1
	0	+∞	+∞	+∞	+∞	+∞	+∞	0
→ **j**	0	1	2	3	4	5	6	↑ **i**

Вычисление DTW

Algorithm DTW $(Q, C \in \mathbb{R}^m)$
$D \in \mathbb{R}^{(1+m)\times(1+m)}, d \in \mathbb{R}^{m\times m}$

$D := \overline{+\infty}$; $D(0,0) := 0$
for $i := 1$ **to** m **do**
  for $j := 1$ to m **do**
    $d(i,j) := \mathrm{Dist}(q_i, c_j)$
    $D(i,j) := d(i,j) + \min\{D(i-1,j), D(i,j-1), D(i-1,j-1)\}$
return $D(m,m)$

Вычисление DTW: сложность (игра не стоит свеч?)

Algorithm DTW $(Q, C \in \mathbb{R}^m)$

$D \in \mathbb{R}^{(1+m)\times(1+m)}, d \in \mathbb{R}^{m\times m}$

$D := \overline{+\infty}; D(0,0) := 0$

for $i := 1$ **to** m **do**

  for $j := 1$ to m **do**

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

    $D(i,j) := d(i,j) + \min\{D(i-1,j), D(i,j-1), D(i-1,j-1)\}$

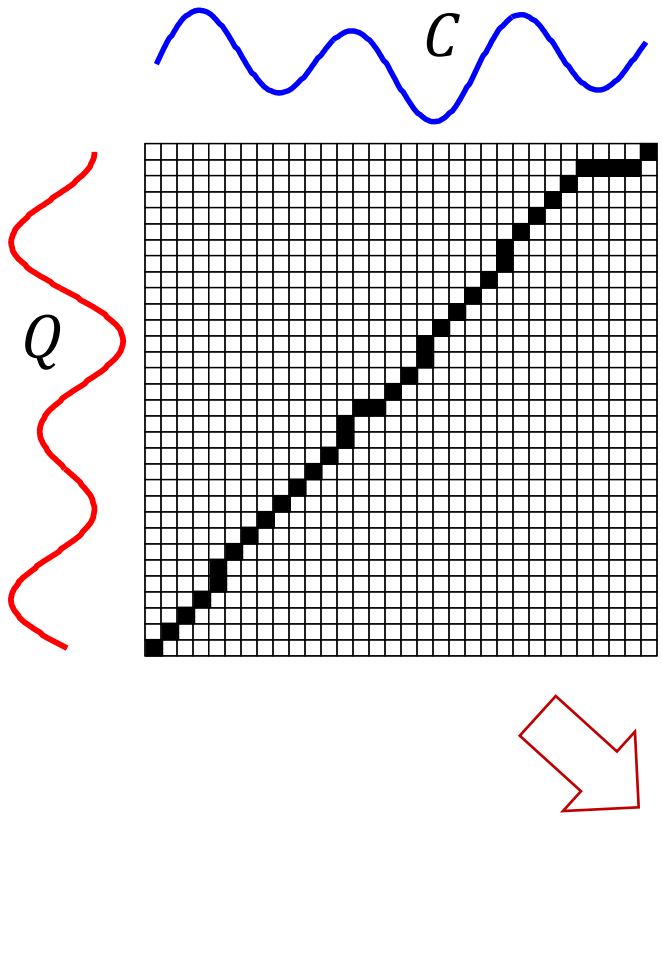
return $D(m,m)$

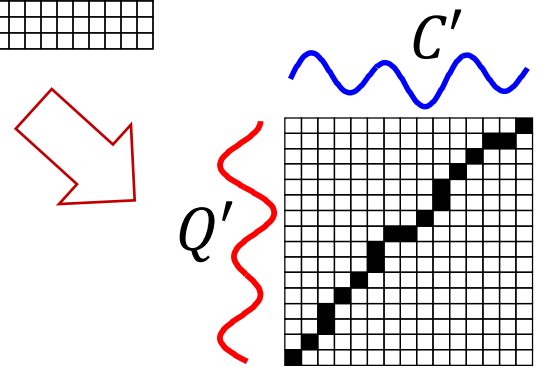
В.М. Васнецов. «Преферанс»

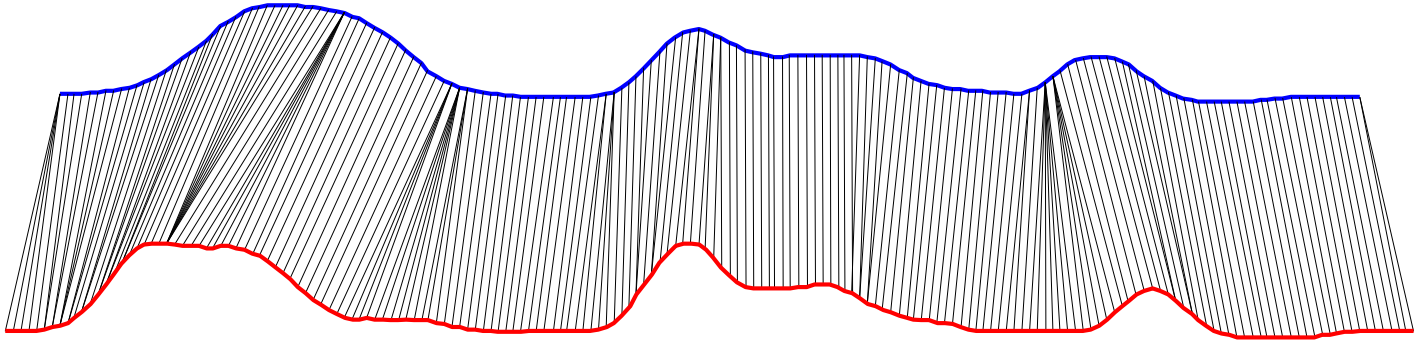
Вычислительная сложность $\boldsymbol{O(m^2)}$

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

Снижение сложности DTW: субдискретизация ряда (downsampling)

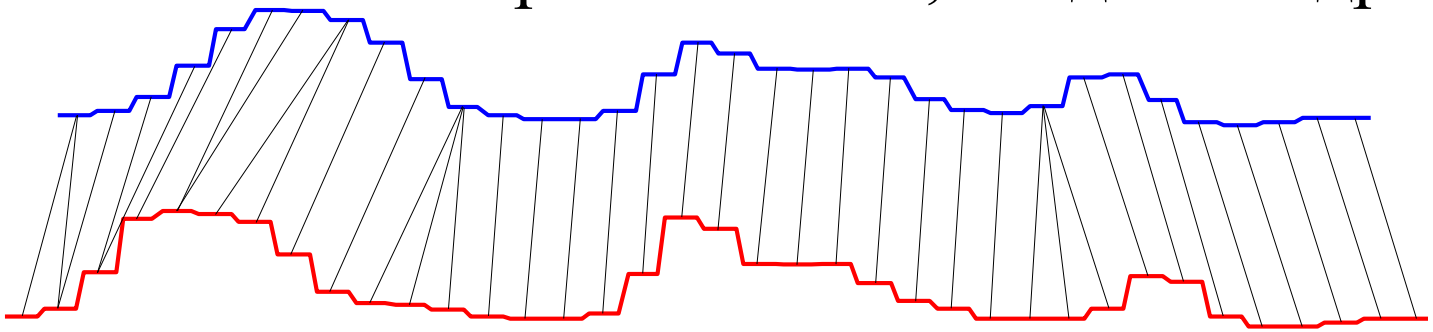






t=23 сек Истинная форма ряда

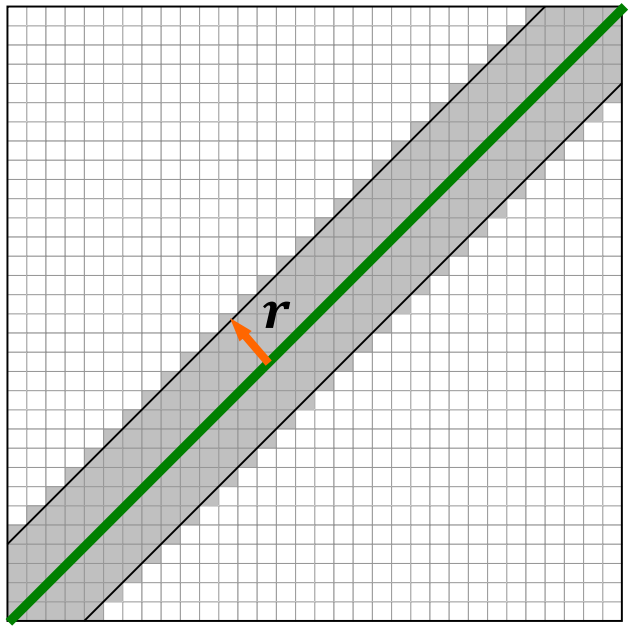
t=1 сек Потерянные пики, впадины и др.



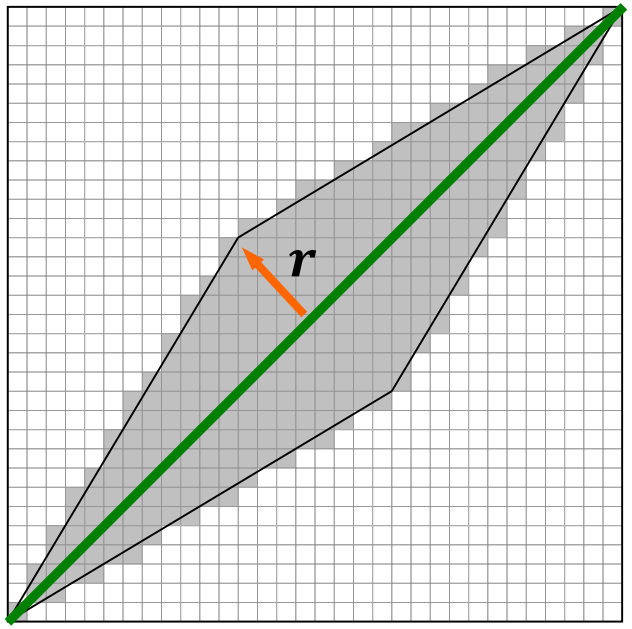
Снижение сложности DTW: ограничение пути трансформации

- Путь не должен отклоняться от диагонали более чем на $\boldsymbol{r}$
- Сложность: $\boldsymbol{O(rm)}$

Полоса Сако–Чиба



Параллелограмм Итакуры



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

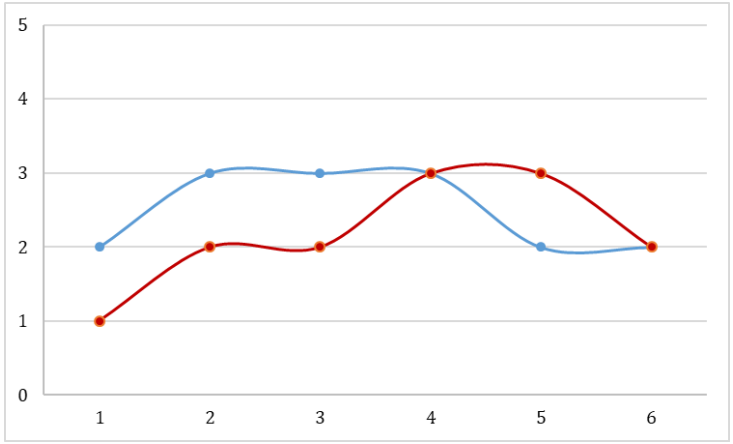
$$D(i,j) = (q_i - c_i)^2 + \min \begin{cases} D(i-1,j) \\ D(i,j-1) \\ D(i-1,j-1) \end{cases}$$

$$D(0,0) = 0, D(i,0) = D(0,j) = +\infty;\ 1 \le i,j \le m;$$

$$\mathbf{0} \le \boldsymbol{r} \le \boldsymbol{m} - \mathbf{1}, j - \boldsymbol{r} \le i \le j + \boldsymbol{r}$$

$$D(i,j) = +\infty, \qquad j + \boldsymbol{r} < i < j - \boldsymbol{r}$$

Пример вычисления DTW с ограничением



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

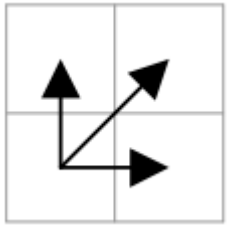
$$D(i,j) = (q_i - c_i)^2 + \min \begin{cases} D(i-1,j) \\ D(i,j-1) \\ D(i-1,j-1) \end{cases}$$

$$D(0,0) = 0, D(i,0) = D(0,j) = +\infty;\ 1 \le i,j \le m;$$

$$\mathbf{0} \le r \le \boldsymbol{m} - \mathbf{1}, j - r \le i \le j + r$$

$$D(i,j) = +\infty, \qquad j + r < i < j - r$$

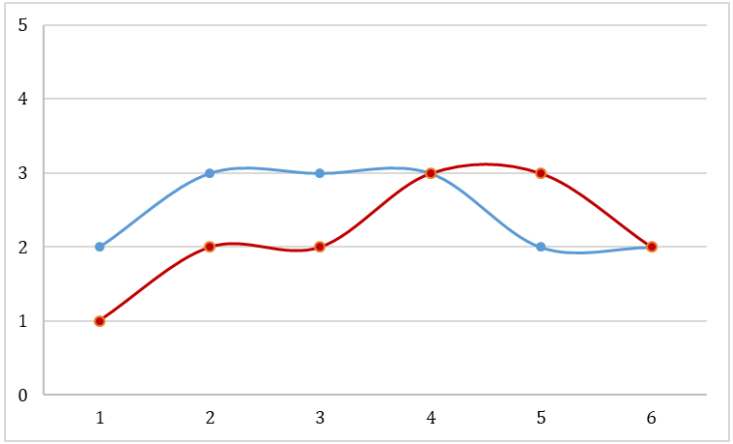
$$C = (2,3,3,3,2,2), Q = (1,2,2,3,3,2)$$



$r = 2$

		2	3	3	3	2	2	
2	+∞	+∞	+∞	+∞	2	1	1	6
3	+∞	+∞	+∞	1	1	2	3	5
3	+∞	+∞	1	1	1	2	3	4
2	+∞	1	2	3	4	4	+∞	3
2	+∞	1	2	3	4	+∞	+∞	2
1	+∞	1	5	9	+∞	+∞	+∞	1
	0	+∞	+∞	+∞	+∞	+∞	+∞	0
→ *j*	0	1	2	3	4	5	6	↑ *i*

Пример вычисления DTW с ограничением



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

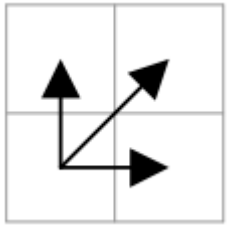
$$D(i,j) = (q_i - c_i)^2 + \min \begin{cases} D(i-1,j) \\ D(i,j-1) \\ D(i-1,j-1) \end{cases}$$

$$D(0,0) = 0, D(i,0) = D(0,j) = +\infty;\ 1 \le i,j \le m;$$

$$\mathbf{0} \le r \le \boldsymbol{m} - \mathbf{1}, j - r \le i \le j + r$$

$$D(i,j) = +\infty, \qquad j + r < i < j - r$$

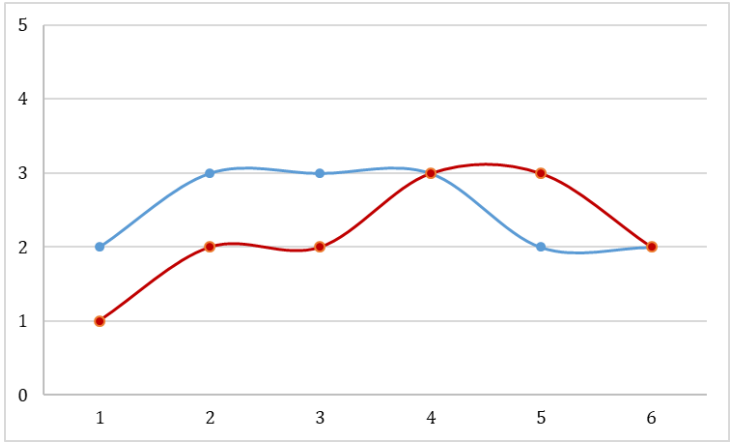
$$C = (2,3,3,3,2,2), Q = (1,2,2,3,3,2)$$



$r = 1$

		2	3	3	3	2	2	
2	+∞	+∞	+∞	+∞	+∞	2	**2**	6
3	+∞	+∞	+∞	+∞	2	3	4	5
3	+∞	+∞	+∞	2	2	3	+∞	4
2	+∞	+∞	2	3	4	+∞	+∞	3
2	+∞	1	2	3	+∞	+∞	+∞	2
1	+∞	1	5	+∞	+∞	+∞	+∞	1
	0	+∞	+∞	+∞	+∞	+∞	+∞	0
→ **j**	0	1	2	3	4	5	6	↑ **i**

Пример вычисления DTW с ограничением



$$\mathbf{DTW}(Q, C) = D(n, n)$$

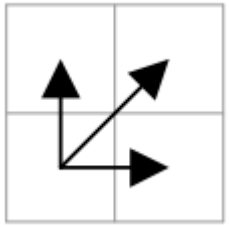
$$D(i,j) = (q_i - c_i)^2 + \min \begin{cases} D(i-1,j) \\ D(i,j-1) \\ D(i-1,j-1) \end{cases}$$

$$D(0,0) = 0, D(i,0) = D(0,j) = +\infty;\ 1 \le i,j \le m;$$

$$\mathbf{0} \le r \le \boldsymbol{m} - \mathbf{1}, j - r \le i \le j + r$$

$$D(i,j) = +\infty, \qquad j + r < i < j - r$$

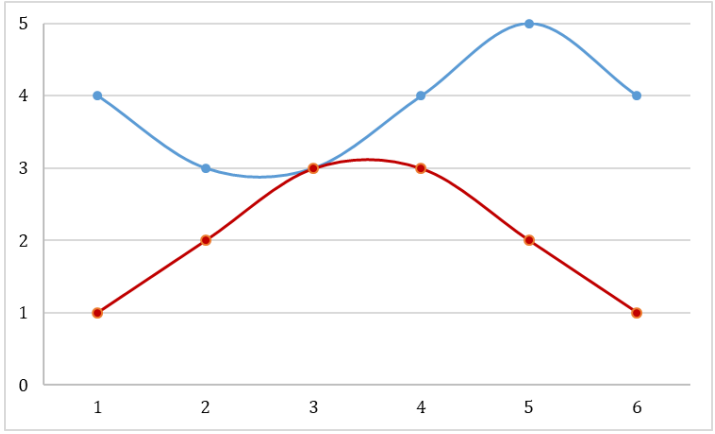
$$C = (2,3,3,3,2,2), Q = (1,2,2,3,3,2)$$



$r = 0$

		2	3	3	3	2	2	
2	+∞	+∞	+∞	+∞	+∞	+∞	**1**	6
3	+∞	+∞	+∞	+∞	+∞	1	+∞	5
3	+∞	+∞	+∞	+∞	1	+∞	+∞	4
2	+∞	+∞	+∞	1	+∞	+∞	+∞	3
2	+∞	+∞	1	+∞	+∞	+∞	+∞	2
1	+∞	1	+∞	+∞	+∞	+∞	+∞	1
	0	+∞	+∞	+∞	+∞	+∞	+∞	0
$\rightarrow j$	0	1	2	3	4	5	6	$\uparrow i$

Пример вычисления DTW с ограничением



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

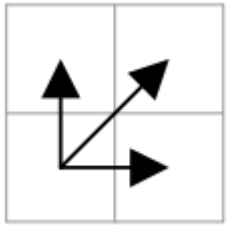
$$D(i,j) = (q_i - c_i)^2 + \min \begin{cases} D(i-1,j) \\ D(i,j-1) \\ D(i-1,j-1) \end{cases}$$

$$D(0,0) = 0, D(i,0) = D(0,j) = +\infty;\ 1 \le i,j \le m;$$

$$\mathbf{0} \le r \le \boldsymbol{m} - \mathbf{1}, j - r \le i \le j + r$$

$$D(i,j) = +\infty, \qquad j + r < i < j - r$$

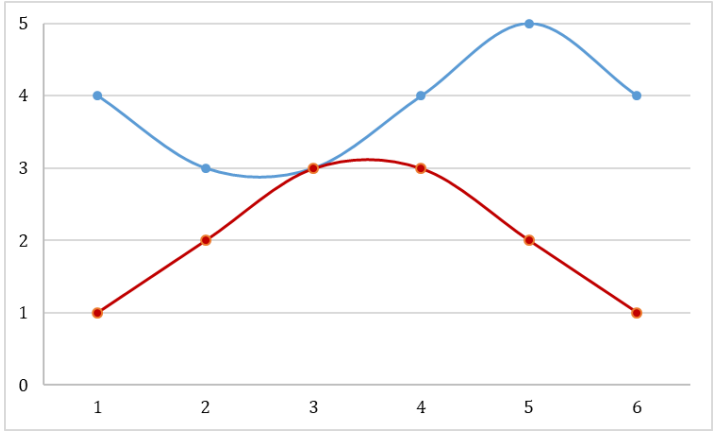
$$C = (4,3,3,4,5,4), Q = (1,2,3,3,2,1)$$



$r = 2$

		4	3	3	4	5	4	
1	+∞	+∞	+∞	+∞	20	30	**28**	6
2	+∞	+∞	+∞	11	14	20	19	5
3	+∞	+∞	10	10	11	15	16	4
3	+∞	14	10	10	11	15	+∞	3
2	+∞	13	10	11	15	+∞	+∞	2
1	+∞	9	13	17	+∞	+∞	+∞	1
	0	+∞	+∞	+∞	+∞	+∞	+∞	0
→ *j*	0	1	2	3	4	5	6	↑ *i*

Пример вычисления DTW с ограничением



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

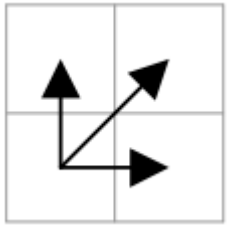
$$D(i,j) = (q_i - c_i)^2 + \min \begin{cases} D(i-1,j) \\ D(i,j-1) \\ D(i-1,j-1) \end{cases}$$

$$D(0,0) = 0, D(i,0) = D(0,j) = +\infty;\ 1 \le i,j \le m;$$

$$\mathbf{0} \le r \le \boldsymbol{m} - \mathbf{1}, j - r \le i \le j + r$$

$$D(i,j) = +\infty, \qquad j + r < i < j - r$$

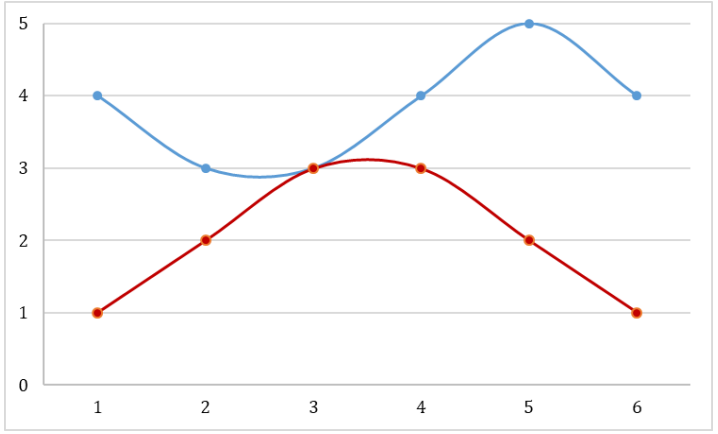
$$C = (4,3,3,4,5,4), Q = (1,2,3,3,2,1)$$



$r = 1$

		4	3	3	4	5	4	
1	+∞	+∞	+∞	+∞	+∞	30	**28**	6
2	+∞	+∞	+∞	+∞	14	20	19	5
3	+∞	+∞	+∞	10	11	15	+∞	4
3	+∞	+∞	10	10	11	+∞	+∞	3
2	+∞	13	10	11	+∞	+∞	+∞	2
1	+∞	9	13	+∞	+∞	+∞	+∞	1
	0	+∞	+∞	+∞	+∞	+∞	+∞	0
→ j	0	1	2	3	4	5	6	↑ i

Пример вычисления DTW с ограничением



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

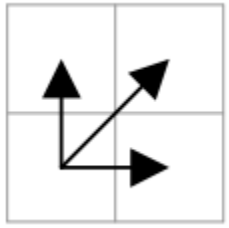
$$D(i,j) = (q_i - c_i)^2 + \min \begin{cases} D(i-1,j) \\ D(i,j-1) \\ D(i-1,j-1) \end{cases}$$

$$D(0,0) = 0, D(i,0) = D(0,j) = +\infty;\ 1 \le i,j \le m;$$

$$\mathbf{0} \le r \le \boldsymbol{m} - \mathbf{1}, j - r \le i \le j + r$$

$$D(i,j) = +\infty, \qquad j + r < i < j - r$$

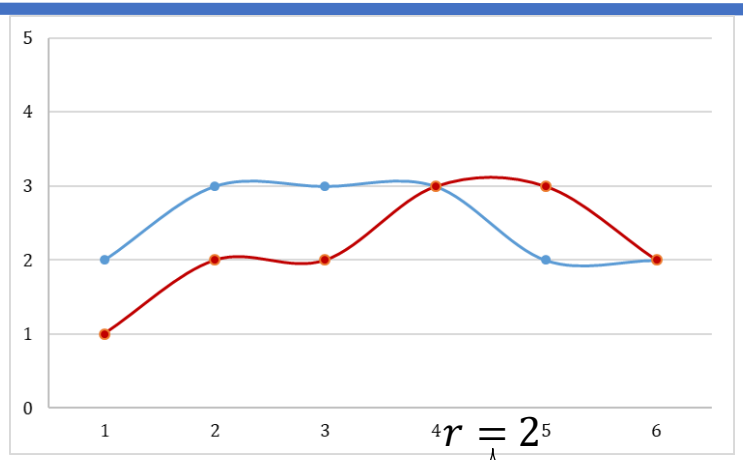
$$C = (4,3,3,4,5,4), Q = (1,2,3,3,2,1)$$



$r = 0$

		4	3	3	4	5	4	
1	+∞	+∞	+∞	+∞	+∞	+∞	**29**	6
2	+∞	+∞	+∞	+∞	+∞	20	+∞	5
3	+∞	+∞	+∞	+∞	11	+∞	+∞	4
3	+∞	+∞	+∞	10	+∞	+∞	+∞	3
2	+∞	+∞	10	+∞	+∞	+∞	+∞	2
1	+∞	9	+∞	+∞	+∞	+∞	+∞	1
	0	+∞	+∞	+∞	+∞	+∞	+∞	0
→ *j*	0	1	2	3	4	5	6	↑ *i*

Сравнение результатов вычисления DTW



		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*

$r = 2$

		2	3	3	3	2	2	
2	+∞	+∞	+∞	+∞	2	1	1	6
3	+∞	+∞	+∞	1	1	2	3	5
3	+∞	+∞	1	1	1	2	3	4
2	+∞	1	2	3	4	4	+∞	3
2	+∞	1	2	3	4	+∞	+∞	2
1	+∞	1	5	9	+∞	+∞	+∞	1
	0	+∞	+∞	+∞	+∞	+∞	+∞	0
→ *j*		0	1	2	3	4	5	6 ↑ *i*

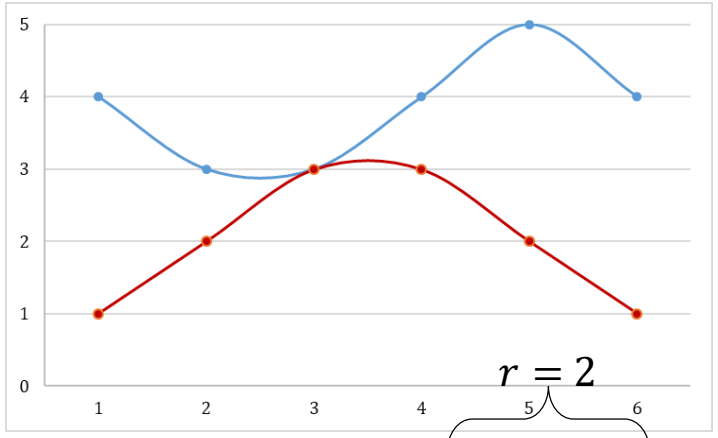
$r = 1$

		2	3	3	3	2	2	
2	+∞	+∞	+∞	+∞	+∞	2	2	6
3	+∞	+∞	+∞	+∞	2	3	4	5
3	+∞	+∞	+∞	2	2	3	+∞	4
2	+∞	+∞	2	3	4	+∞	+∞	3
2	+∞	1	2	3	+∞	+∞	+∞	2
1	+∞	1	5	+∞	+∞	+∞	+∞	1
	0	+∞	+∞	+∞	+∞	+∞	+∞	0
→ *j*		0	1	2	3	4	5	6 ↑ *i*

$r = 0$

		2	3	3	3	2	2	
2	+∞	+∞	+∞	+∞	+∞	+∞	1	6
3	+∞	+∞	+∞	+∞	+∞	1	+∞	5
3	+∞	+∞	+∞	+∞	1	+∞	+∞	4
2	+∞	+∞	+∞	1	+∞	+∞	+∞	3
2	+∞	+∞	1	+∞	+∞	+∞	+∞	2
1	+∞	1	+∞	+∞	+∞	+∞	+∞	1
	0	+∞	+∞	+∞	+∞	+∞	+∞	0
→ *j*		0	1	2	3	4	5	6 ↑ *i*

Сравнение результатов вычисления DTW



		4	3	3	4	5	4	
1	+∞	+∞	+∞	+∞	20	30	**28**	6
2	+∞	+∞	+∞	11	14	20	19	5
3	+∞	+∞	10	10	11	15	16	4
3	+∞	14	10	10	11	15	+∞	3
2	+∞	13	10	11	15	+∞	+∞	2
1	+∞	9	13	17	+∞	+∞	+∞	1
	0	+∞	+∞	+∞	+∞	+∞	+∞	0
$\rightarrow j$	0	1	2	3	4	5	6	$\uparrow i$

$r = 1$

		4	3	3	4	5	4	
1	+∞	+∞	+∞	+∞	+∞	30	**28**	6
2	+∞	+∞	+∞	+∞	14	20	19	5
3	+∞	+∞	+∞	10	11	15	+∞	4
3	+∞	+∞	10	10	11	+∞	+∞	3
2	+∞	13	10	11	+∞	+∞	+∞	2
1	+∞	9	13	+∞	+∞	+∞	+∞	1
	0	+∞	+∞	+∞	+∞	+∞	+∞	0
$\rightarrow j$	0	1	2	3	4	5	6	$\uparrow i$

$r = 0$

		4	3	3	4	5	4	
1	+∞	+∞	+∞	+∞	+∞	+∞	**29**	6
2	+∞	+∞	+∞	+∞	+∞	20	+∞	5
3	+∞	+∞	+∞	+∞	11	+∞	+∞	4
3	+∞	+∞	+∞	10	+∞	+∞	+∞	3
2	+∞	+∞	10	+∞	+∞	+∞	+∞	2
1	+∞	9	+∞	+∞	+∞	+∞	+∞	1
	0	+∞	+∞	+∞	+∞	+∞	+∞	0
$\rightarrow j$	0	1	2	3	4	5	6	$\uparrow i$

Вычисление DTW с ограничением

Algorithm DTW $(Q, C \in \mathbb{R}^m$, **int** $r)$

$D \in \mathbb{R}^{(1+m)\times(1+m)}, d \in \mathbb{R}^{m\times m}$

$D := \overline{+\infty}; D(0,0) := 0$

for $i := 1$ **to** m **do**

  for $j := \max(1, i-r)$ **to** $\min(m, i+r)$ **do**

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

    $D(i,j) := d(i,j) + \min\{D(i-1,j), D(i,j-1), D(i-1,j-1)\}$

return $D(m,m)$

Вычисление DTW с ограничением: сложность

Algorithm DTW $(Q, C \in \mathbb{R}^m$, **int** $r)$

$D \in \mathbb{R}^{(1+m)\times(1+m)}, d \in \mathbb{R}^{m\times m}$

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

$D := \overline{+\infty}; D(0,0) := 0$

for $i := 1$ **to** m **do**

  for $j := \max(1, i-r)$ **to** $\min(m, i+r)$ **do**

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

    $D(i,j) := d(i,j) + \min\{D(i-1,j), D(i,j-1), D(i-1,j-1)\}$

return $D(m,m)$

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

Дж.У. Хенс. «Козырь»

Содержание

Сложность DTW снижена, но как ускорить поиск по образцу?

Algorithm NaïveSearch ($Q \in \mathbb{R}^m, T, r$)

$bsf := +\infty$
for each $C_i \in S_T^m$ **do**
  $dist := \mathrm{DTW}(Q, C_i, r)$
  if $dist < bsf$ **then**
    $bsf := dist$
    $C_{\mathrm{bestmatch}} := C_i$
return $C_{\mathrm{bestmatch}}$

- Неравенство треугольника не работает, $\mathrm{DTW}(Q, C_i)$ нельзя применить в качестве нижней границы!
- Вычислительная сложность $O(nmr)$

Нижняя граница (lower bound)

Algorithm LBsearch $(Q \in \mathbb{R}^m, T, r)$

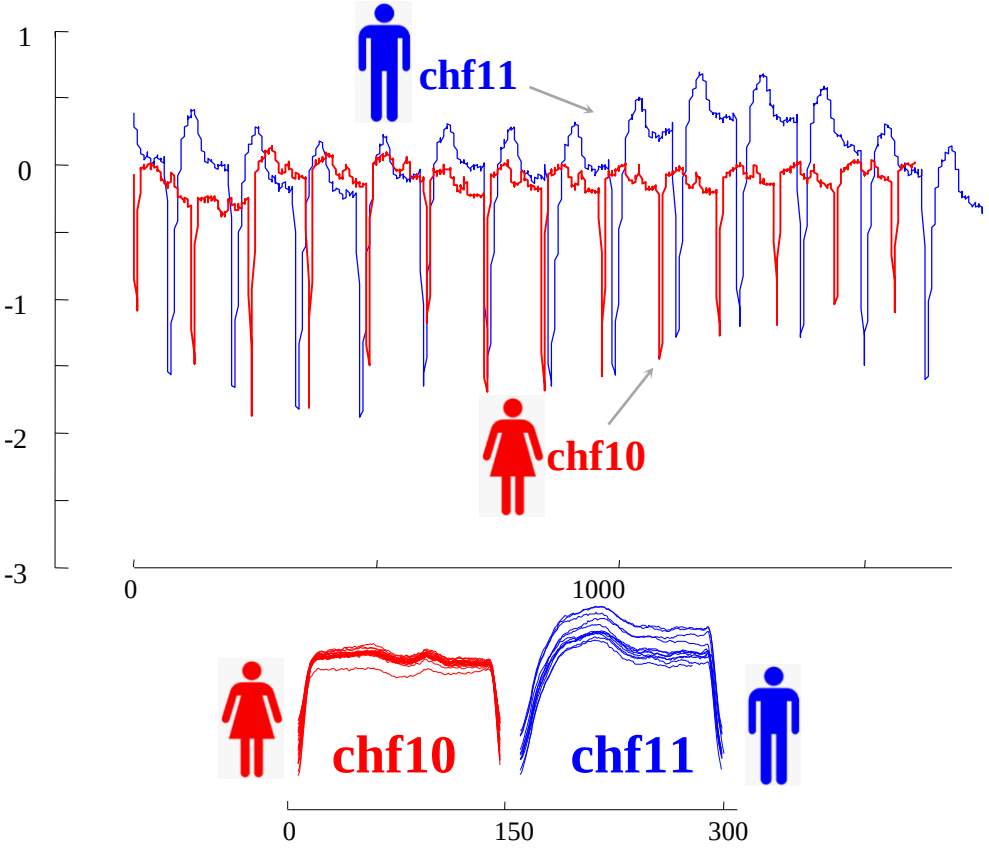
$bsf := +\infty$
for each $C_i \in S_T^m$ **do**
  if $\mathbf{LB}(Q, C_i) < bsf$ **then**
    $dist := \mathrm{DTW}(Q, C_i, r)$
    if $dist < bsf$ **then**
      $bsf := dist$
      $C_{\mathrm{bestmatch}} := C_i$
return $C_{\mathrm{bestmatch}}$

- Функция **LB**: $\mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}_+ \cup \{0\}$ со сложностью меньше $O(m^2)$

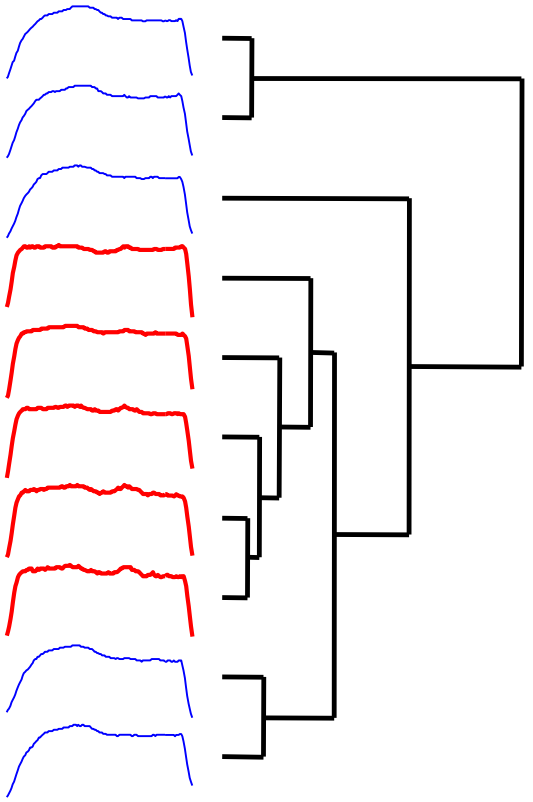
  $\forall\, C, Q$: $\mathbf{LB}(Q, C) \le \mathrm{DTW}(Q, C)$
- Если $\mathbf{LB}(Q, C_i) > bsf$, то $\mathrm{DTW}(Q, C_i) > bsf$, т.е. C_i заведомо непохож на Q и не нужно вычислять $\mathrm{DTW}(Q, C_i)$
- Необходима **z-нормализация S_T^m и Q (всех подпоследовательностей** и запроса)

Важность z-нормализации подпоследовательностей

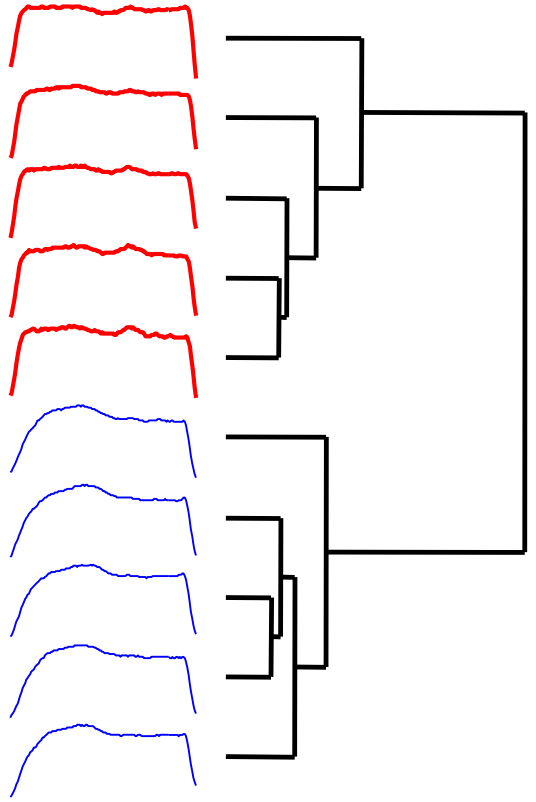
BIDMC Congestive Heart Failure Database*



Не нормализованные

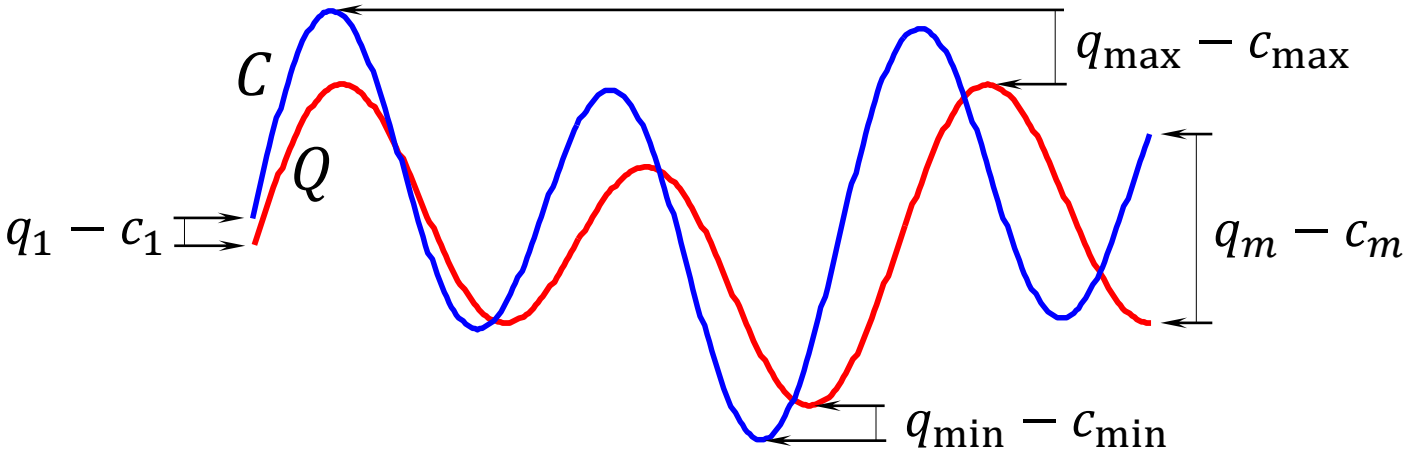


Z-нормализованные



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

Нижние границы $\mathrm{LB_{Kim}}$ и $\mathrm{LB_{Kim}FL}^*$



Санг-Вук Ким
(Sang-Wook Kim)

- Квадрат разности между парой точек Q и C:
  - первая и последняя, сложность $O(1)$:

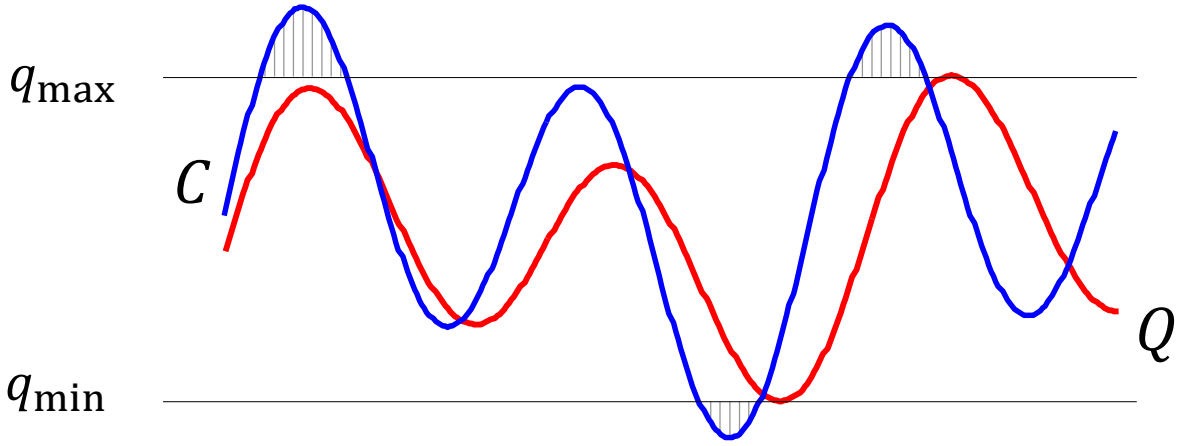
$$\mathrm{LB_{Kim}FL}(Q,C) = (q_1 - c_1)^2 + (q_m - c_m)^2$$

  - минимум и максимум, сложность $O(m)$:

$$\mathrm{LB_{Kim}}(Q,C) = (q_{\max} - c_{\max})^2 + (q_{\min} - c_{\min})^2$$

*Kim S., *et al.* An index-based approach for similarity search supporting time warping in large sequence databases. Proc. of the 17th Int. Conf. on Data Engineering, ICDE 01, April 2-6, 2001, Heidelberg, Germany, pp. 607-614. DOI: 10.1109/ICDE.2001.914875

Нижняя граница LB_{Yi}*



Байоюнг-Ки Йи
(Byoung-Kee Yi)

Сумма квадратов длин ||| дает мин. вклад в DTW, сложность $O(m)$:

$$\mathrm{LB_{Yi}}(Q,C) = \sum_{c_i > \max(q_1,\ldots,q_m)} c_i^2 + \sum_{c_i < \min(q_1,\ldots,q_m)} c_i^2$$

* Yi B., Jagadish H., Faloutsos C. Efficient retrieval of similar time sequences under time warping. Proc. of the 14th Int. Conf. on Data Engineering, ICDE 98, Orlando, Florida, USA, February 23-27, 1998, pp. 23-27. DOI: 10.1109/ICDE.1998.655778

Нижняя граница $\mathbf{LB_{Keogh}}$*

$$\mathbf{LB_{Keogh}}(Q,C) = \sum_{i=1}^{m} \begin{cases} (c_i - u_i)^2, c_i > u_i \\ (c_i - \ell_i)^2, c_i < \ell_i \\ 0, \quad otherwise \end{cases}$$

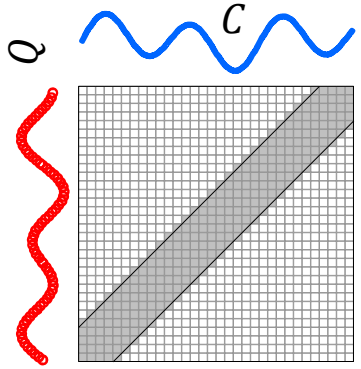
Верхняя оболочка U

$$u_i = \max_{i-r \le k \le i+r} q_k$$

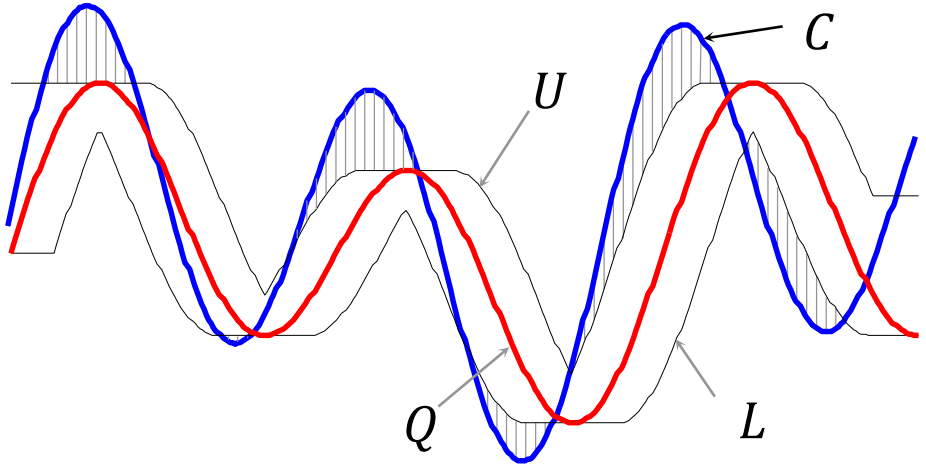
Нижняя оболочка L

$$\ell_i = \min_{i-r \le k \le i+r} q_k$$

Имонн Кеог
(Eamonn Keogh)



Полоса Сако—Чиба



NB! Для DTW (*не квадрата* DTW)

$$\mathbf{LB_{Keogh}}(Q,C) = \sqrt{\sum_{i=1}^{m} \begin{cases} (c_i - u_i)^2, c_i > u_i \\ (c_i - \ell_i)^2, c_i < \ell_i \\ 0, \quad otherwise \end{cases}}$$

NB! $\mathbf{LB_{Keogh}}$ работает только в случае рядов равной длины ($|Q| = |C|$)

* Keogh E.J. Exact Indexing of Dynamic Time Warping. VLDB 2002. pp. 406–417. DOI: 10.1016/B978-155860869-6/50043-3

Нижняя граница $\mathbf{LB_{Keogh}}$*

$$\mathbf{LB_{Keogh}}(Q,C) = \sum_{i=1}^{m} \begin{cases} (c_i - u_i)^2, c_i > u_i \\ (c_i - \ell_i)^2, c_i < \ell_i \\ 0, \quad otherwise \end{cases}$$

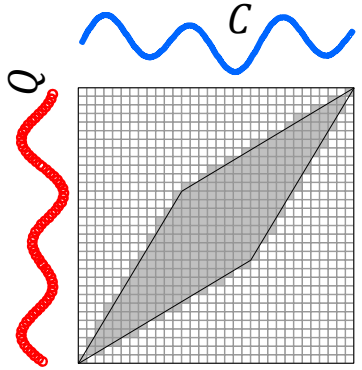
Верхняя оболочка U

$$u_i = \max_{i-r \le k \le i+r} q_k$$

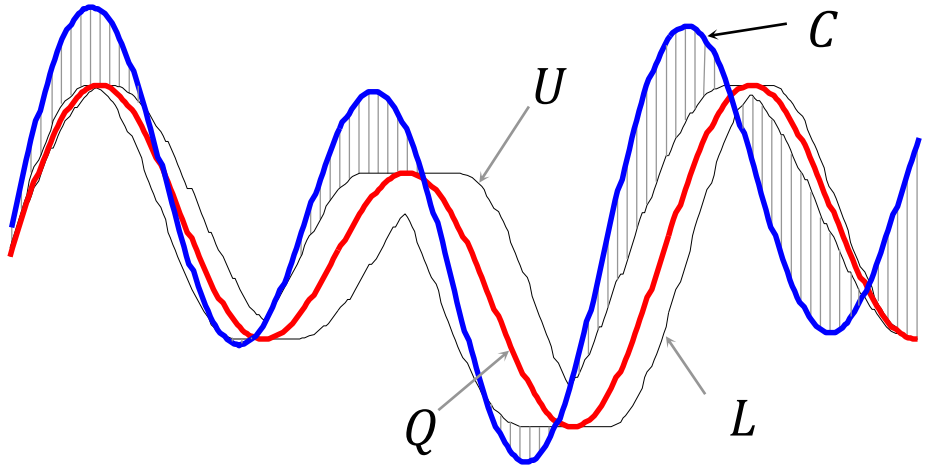
Нижняя оболочка L

$$\ell_i = \min_{i-r \le k \le i+r} q_k$$

Имонн Кеог
(Eamonn Keogh)



Параллелограмм Итакуры



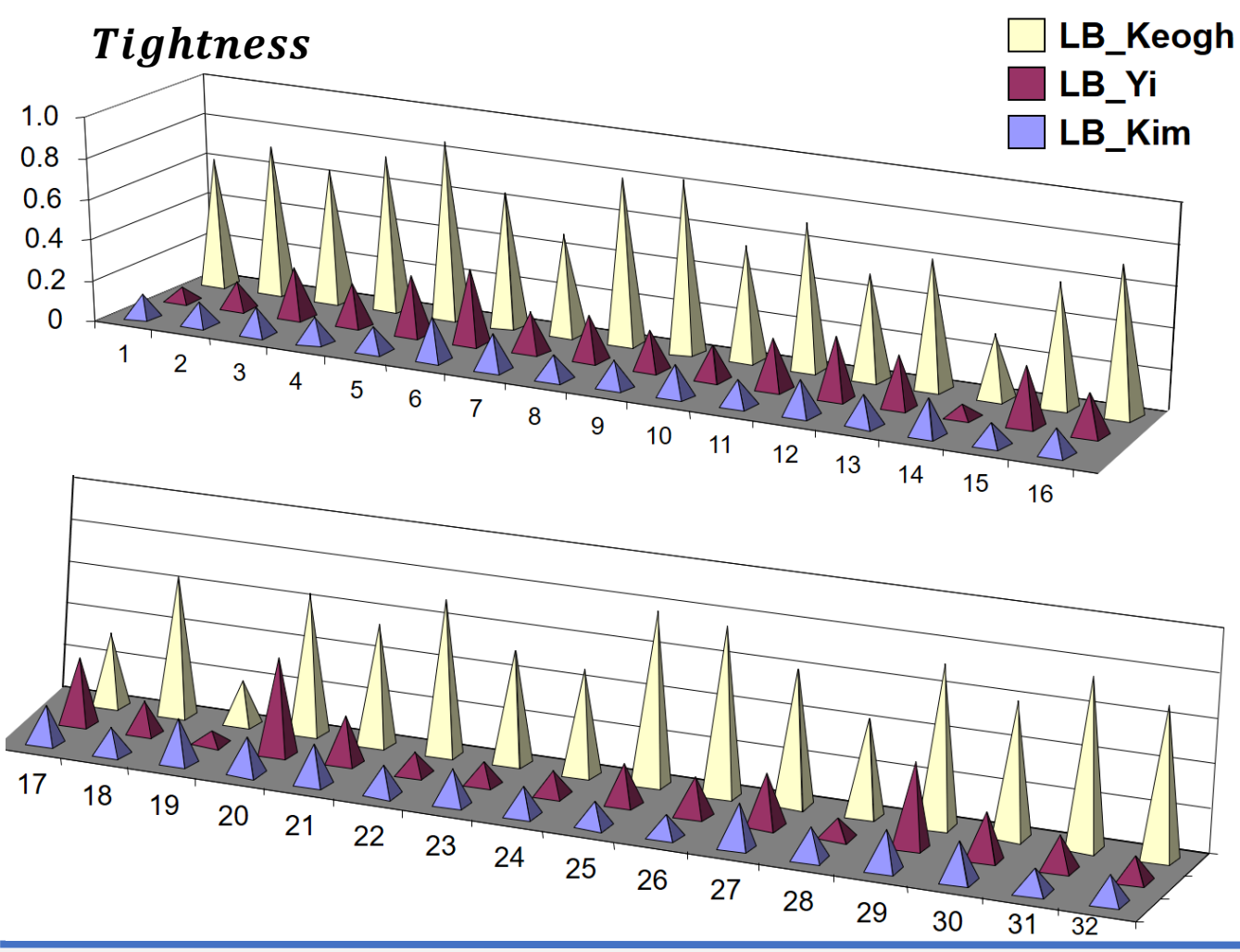
NB! Для DTW (*не квадрата* DTW)

$$\mathbf{LB_{Keogh}}(Q,C) = \sqrt{\sum_{i=1}^{m} \begin{cases} (c_i - u_i)^2, c_i > u_i \\ (c_i - \ell_i)^2, c_i < \ell_i \\ 0, \quad otherwise \end{cases}}$$

NB! $\mathbf{LB_{Keogh}}$ работает только в случае рядов равной длины ($|Q| = |C|$)

* Keogh E.J. Exact Indexing of Dynamic Time Warping. VLDB 2002. pp. 406–417. DOI: 10.1016/B978-155860869-6/50043-3

Сравнение нижних границ: Tightness (узость)

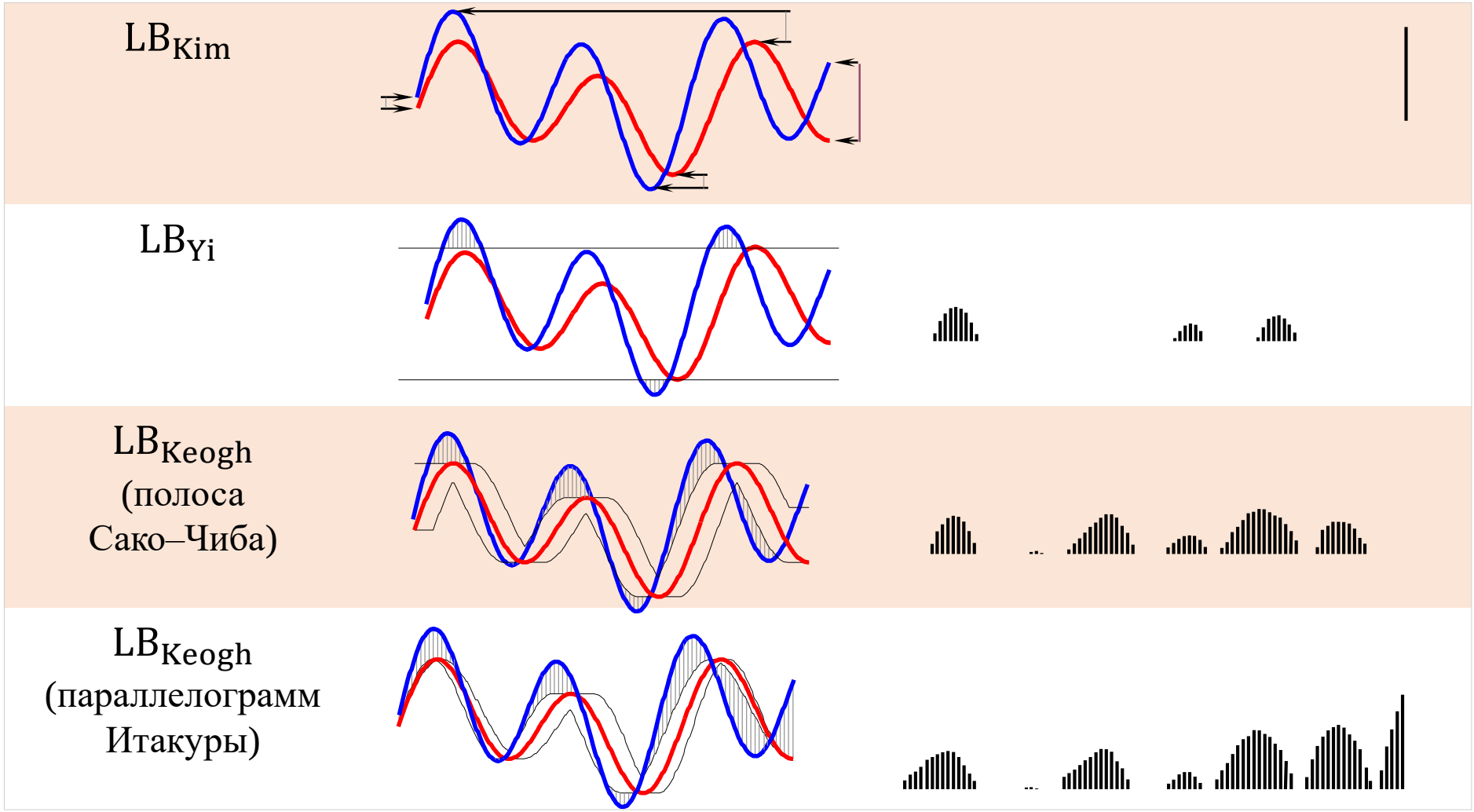


- Взяли 32 временных ряда из различных предметных областей
- В каждом ряде взяли 50 случайных подпоследовательностей наиболее типичной длины 256
- Для каждой пары подпоследовательностей Q, C вычислили $\mathrm{LB}(Q, C)$ и $\mathrm{DTW}(Q, C)$ и усреднили
- В итоге вычислили метрику

$$\boldsymbol{Tightness = \frac{LB}{DTW}}$$

($0 \leq Tightness \leq 1$, большее значение лучше)

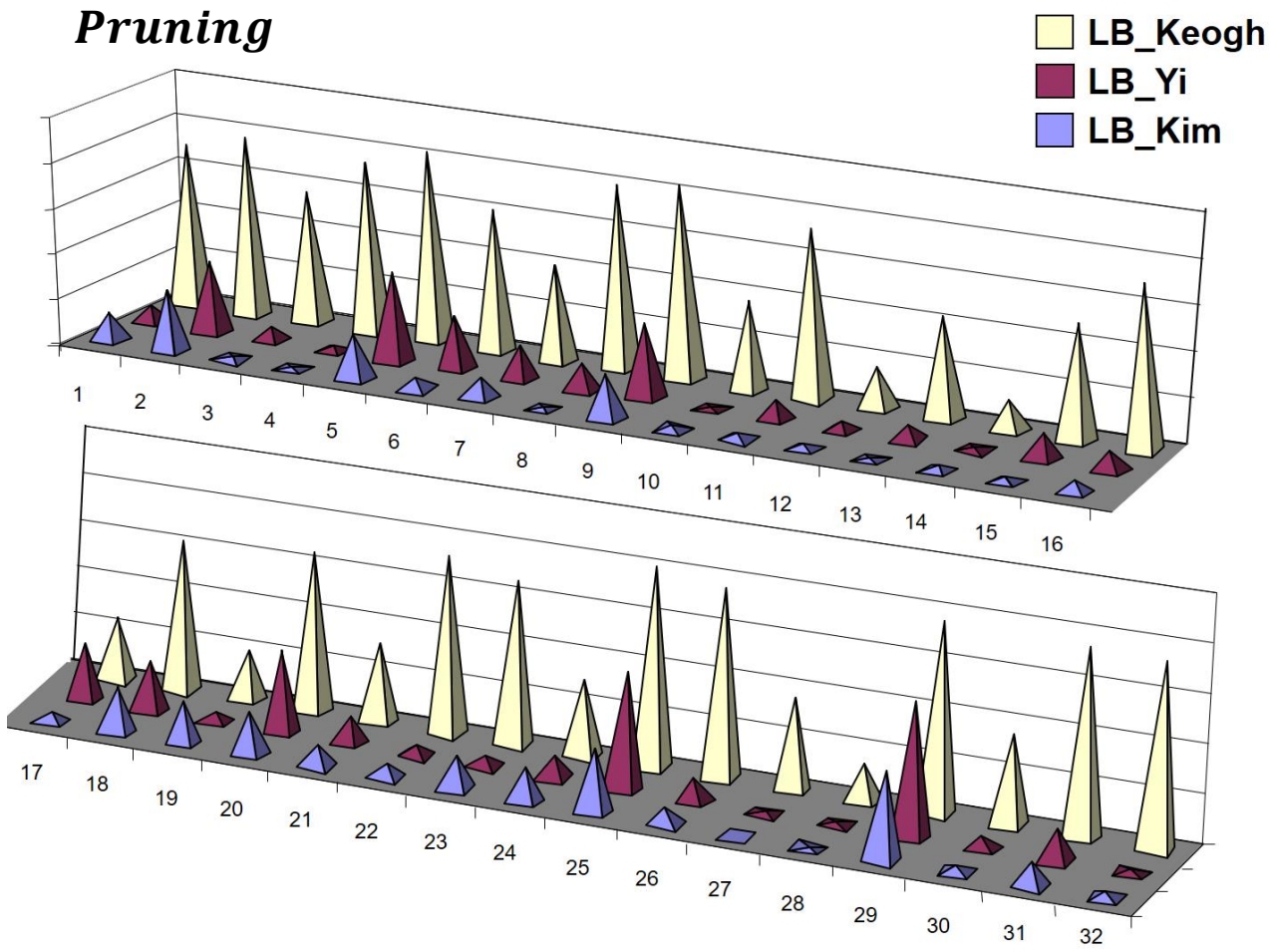
Сравнение нижних границ: Tightness (узость)



Tightness

нижней границы пропорциональна длине |||

Сравнение нижних границ: Pruning power (объем отбрасывания)



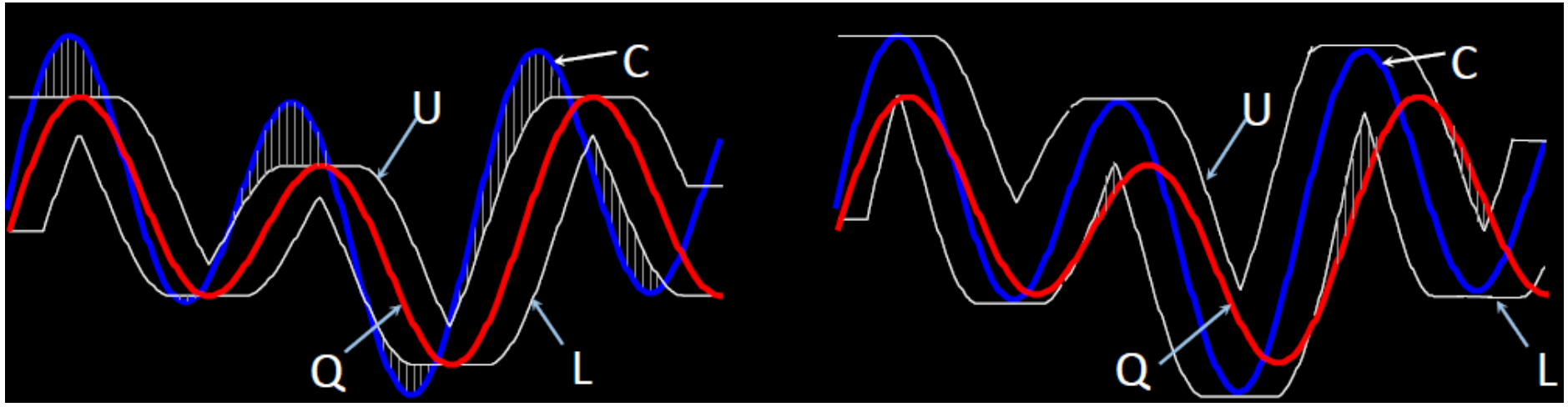
- Взяли 32 временных ряда из различных предметных областей
- В каждом ряде взяли 50 случайных подпоследовательностей наиболее типичной длины 256
- Каждую подпоследовательность взяли как запрос для поиска наиболее похожей подпоследовательности среди остальных и усреднили число случаев где НЕ вычисляли DTW
- В итоге вычисли метрику

$$\boldsymbol{Pruning} = \frac{\text{число случаев где НЕ вычисляли } \mathbf{DTW}}{|S_T^m|}$$

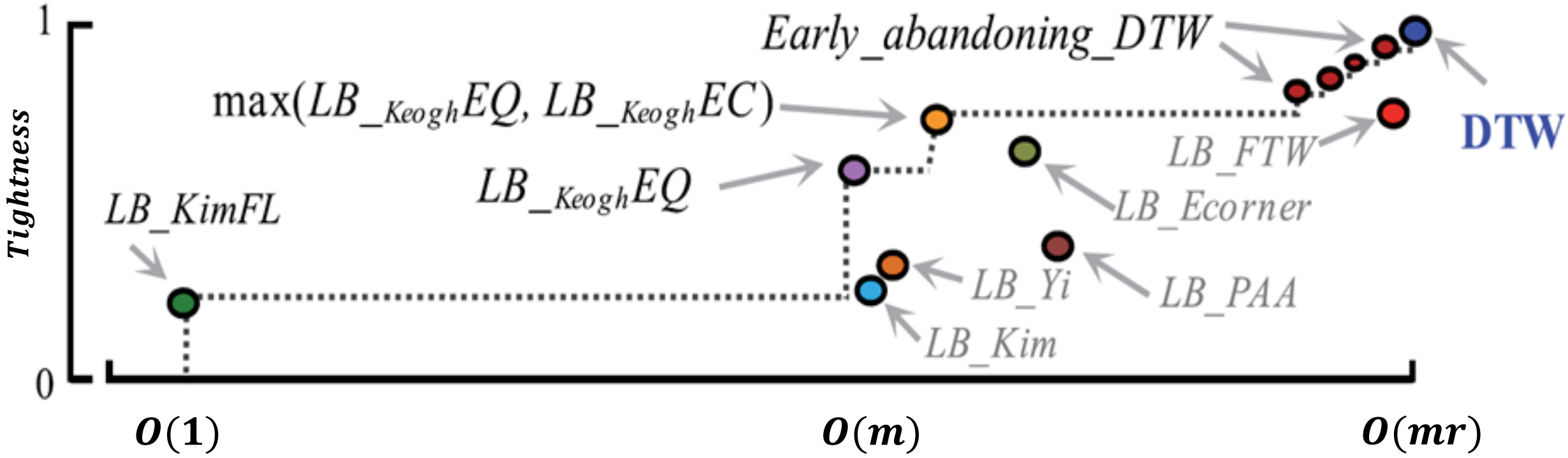
($0 \le Pruning \le 1$,
большее значение лучше)

Нижняя граница $\mathbf{LB_{Keogh}EC}$

- Оболочка строится вокруг запроса (аббр. Envelope around the Query): нижняя граница $\mathrm{LB_{Keogh}EQ} \equiv \mathrm{LB_{Keogh}}$
- Q и C меняются местами (аббр. Envelope around the Candidate): нижняя граница $\mathbf{LB_{Keogh}EC}(Q, C) = \mathrm{LB_{Keogh}EQ}(C, Q)$
  NB! В общем случае $\mathrm{LB_{Keogh}EC} \neq \mathrm{LB_{Keogh}EQ}$



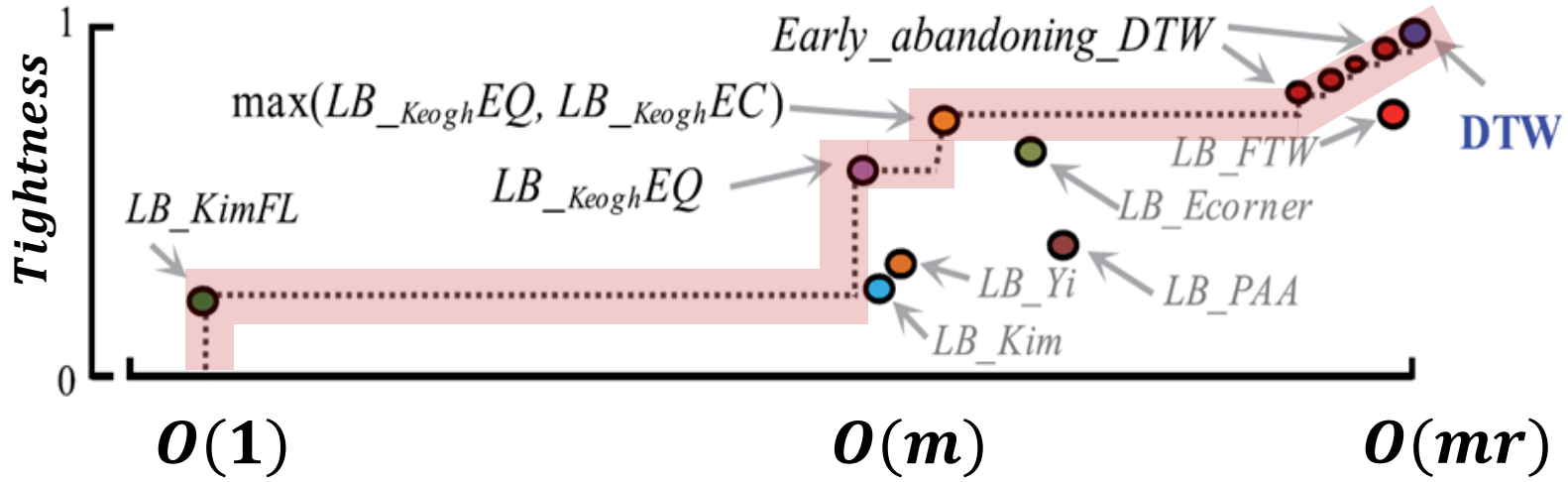
Прочие нижние границы* (18+)



* Rakthanmanon T. *et al.* Addressing big data time series: Mining trillions of time series subsequences under Dynamic Time Warping. TKDD. 2013. Vol. 7, No. 3. P. 10. DOI: 10.1145/2500489

Какие нижние границы применять?

- Все в линейке ····· каскадом от менее к более вычислительно затратным
- Отказ от любой нижней границы в каскаде замедляет вычисления минимум вдвое
- Применение каскада позволяет достичь $\boldsymbol{Prune = 0.99999}$*



Большой каскад / Гос. музей-заповедник «Петергоф»

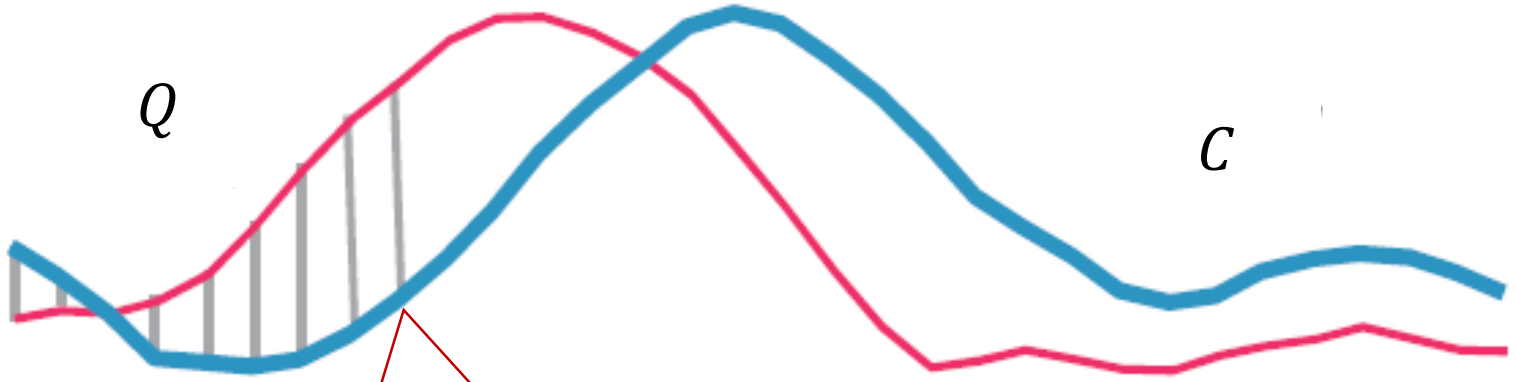
* Rakthanmanon T. *et al.* Addressing big data time series: Mining trillions of time series subsequences under Dynamic Time Warping. TKDD. 2013. Vol. 7, No. 3. P. 10. DOI: 10.1145/2500489

Оптимизация: ранний останов вычислений (early abandoning)

Прекратить вычисления, если текущий результат больше bsf

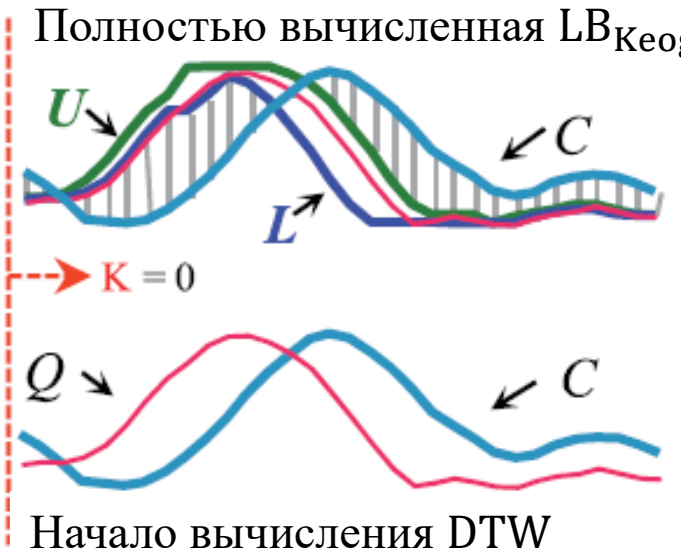
1. Ранний останов вычисления ED и LB_{Keogh}
2. Ранний останов вычисления DTW
3. Более раннее отбрасывание DTW с помощью LB_{Keogh}
4. Переупорядочивание раннего отбрасывания

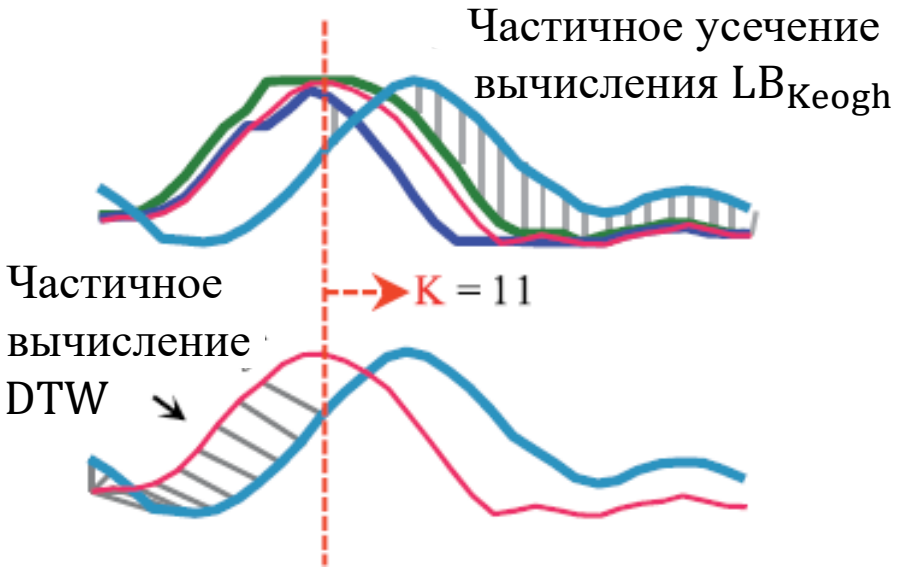
Ранний останов вычисления ED и LB_{Keogh}



На этой точке можно остановить вычисления, если текущий результат $\sum(q_i - c_i)^2$ превысил bsf

Ранний останов вычисления DTW

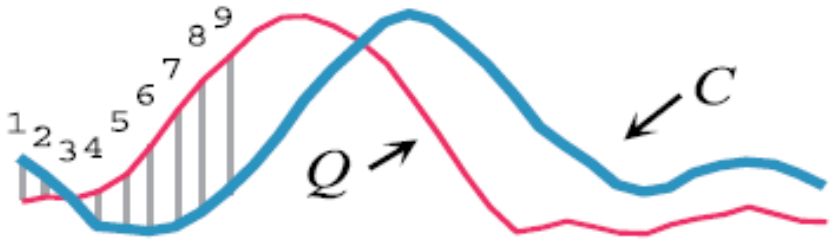




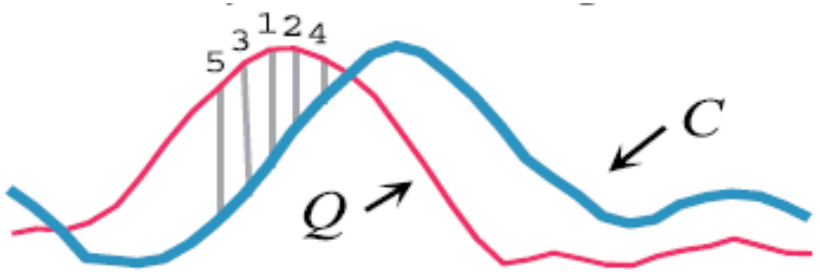
- Пусть LB_{Keogh} подсчитана, но необходимо вычислить DTW. Можно выполнять инкрементное вычисление DTW слева направо и по мере постепенного вычисления от 1 до K суммировать частичное накопление DTW с вкладом LB_{Keogh} от $K + 1$ до m
- $\min_j \text{DTW}(Q_{1:K}, C_{1:K+j}) + \text{LB}_{\text{Keogh}}(Q_{K+r+1:m}, C_{K+r+1:m}) < DTW(Q_{1:m}, C_{1:m})$

Переупорядочивание вычислений (reordering) и их ранний останов

Обычный порядок вычислений
до их раннего останова



Оптимизированный порядок
вычислений до их раннего останова



- Эвристика: оптимальным упорядочением является сортировка индексов на основе абсолютных значений $\hat{Q}$. Обоснование: значение q_i будет сравниваться со многими значениями c_i во время поиска. Распределение многих c_i будет гауссовым, со средним значением 0. Поэтому промежутки из Q, наиболее удаленные от μ_Q, в среднем вносят наибольший вклад в расстояние
- Проверка эвристики: для подпоследовательности ряда ЭКГ вычислили полное евклидово расстояние до 10^6 других случайных подпоследовательностей ЭКГ. Затем нашли наилучший порядок вычислений: взяли c_i и отсортировали их по убыванию их вкладов в евклидово расстояние. В итоге сравнили это эмпирически оптимальное упорядочение с прогнозируемым (сортировка индексов по абсолютным значениям Q) и обнаружили, что ранговая корреляция составляет 0.999
- Прием подходит для ED, $\mathrm{LB}_{\mathrm{Keogh}}$ и может сочетаться с ранним остановом Z-нормализации

Литература

1. Zhong S., Mueen A. MASS: distance profile of a query over a time series. Data Mining and Knowledge Discovery. 2024. Vol. 38. P. 1466–1492. https://doi.org/10.1007/s10618-024-01005-2.
2. Rakthanmanon T., Campana B.J.L., Mueen A. *et al.* Addressing big data time series: Mining trillions of time series subsequences under Dynamic Time Warping. TKDD. 2013. Vol. 7, No. 3. P. 10. https://doi.org/10.1145/2500489.
3. Ratanamahatana C.A., Keogh E.J. Three myths about Dynamic Time Warping data mining. Proc. of the 2005 SIAM Int. Conf. on Data Mining, SDM 2005, Newport Beach, CA, USA, April 21–23, 2005. 2005. P. 506–510. https://doi.org/10.1137/1.9781611972757.50.
4. Sakoe H., Chiba S. Dynamic programming algorithm optimization for spoken word recognition. IEEE Trans. on Acoustics, Speech, and Signal Processing, 1978. 26(1), 43-49. https://doi.org/10.1109/TASSP.1978.1163055