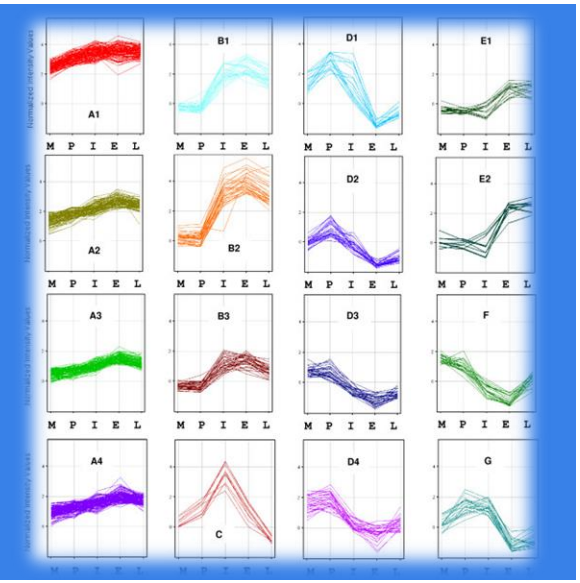
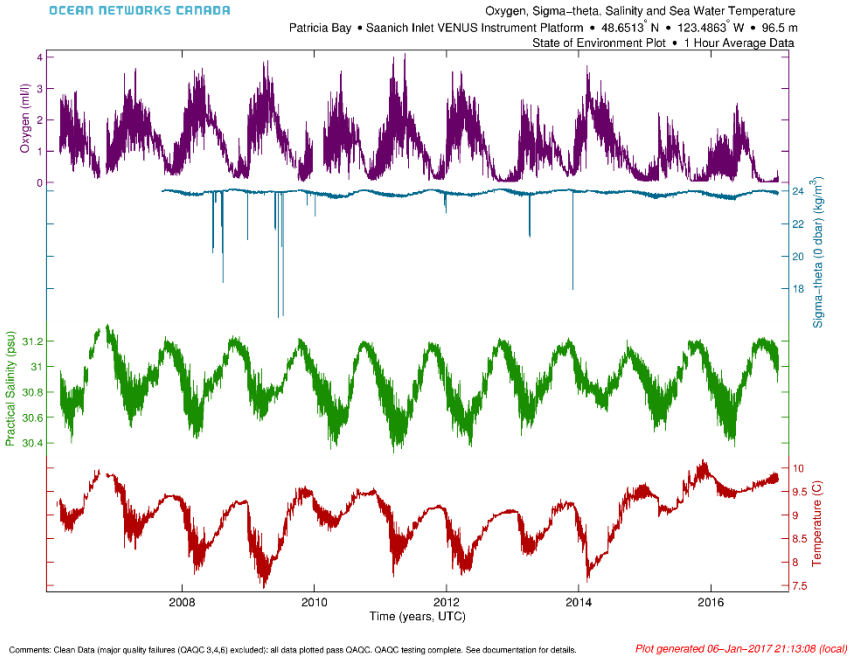


Time Series Clustering

목차

시계열 데이터

- 센서 기술 발전으로 다양한 분야에서 실시간으로 데이터 수집이 가능
  → 공정 설비, 건설 장비 등..
- 시계열(time series) 데이터는 관측치가 시간적 순서를 갖음
  → "시간"을 고려한 분석이 필요



시계열 데이터 특징

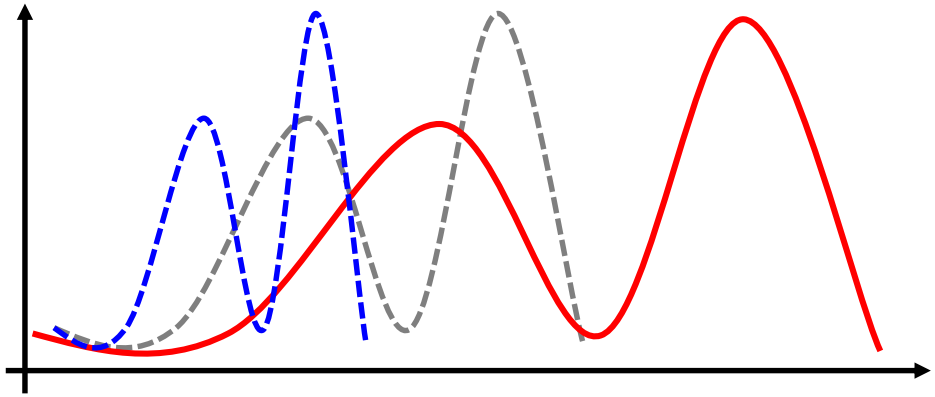
- 시계열 데이터는 다음과 같은 특징이 있음

1. Scaling and translation invariance
2. Shift invariance
3. Uniform scaling invariance
4. Complexity invariance

시계열 데이터 특징

- 시계열 데이터는 다음과 같은 특징이 있음

1. **Scaling and translation invariance**

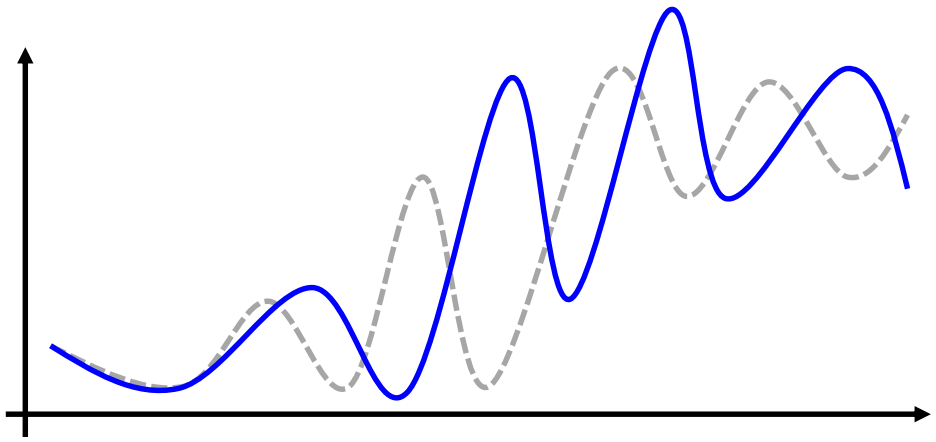


2. Shift invariance
3. Uniform scaling invariance
4. Complexity invariance

시계열 데이터 특징

- 시계열 데이터는 다음과 같은 특징이 있음

1. Scaling and translation invariance
2. **Shift invariance (misalignment)**

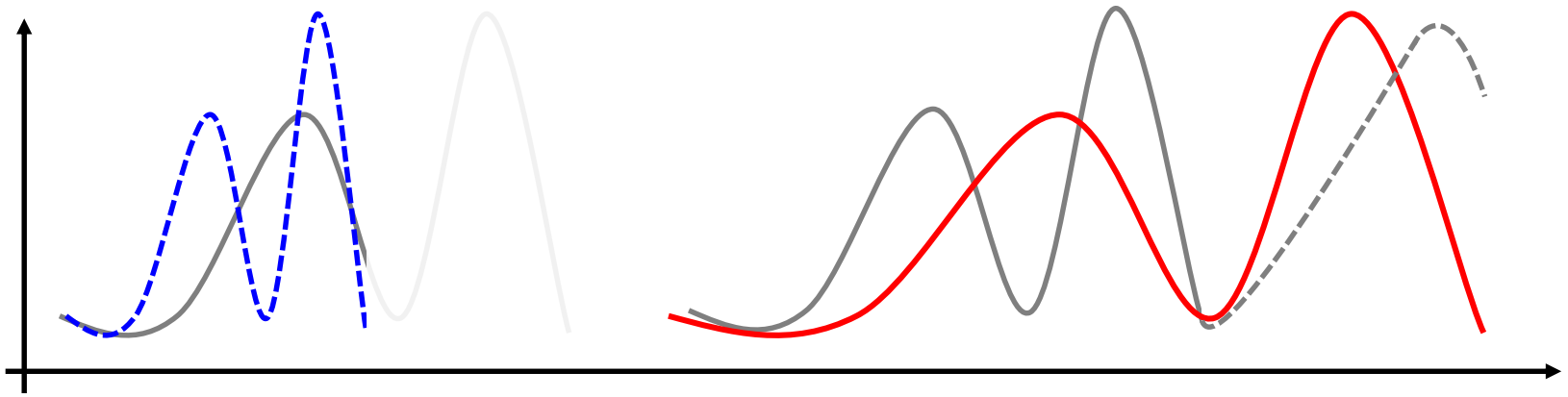


3. Uniform scaling invariance
4. Complexity invariance

시계열 데이터 특징

- 시계열 데이터는 다음과 같은 특징이 있음

1. Scaling and translation invariance
2. Shift invariance
3. **Uniform scaling invariance**

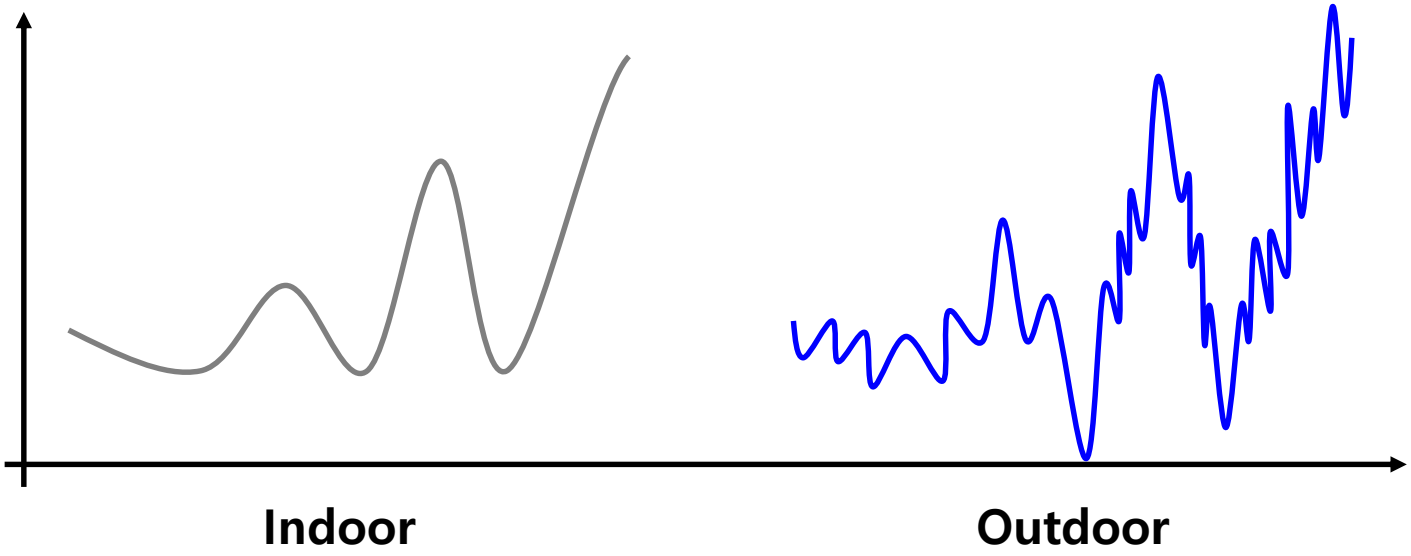


4. Complexity invariance

시계열 데이터 특징

- 시계열 데이터는 다음과 같은 특징이 있음

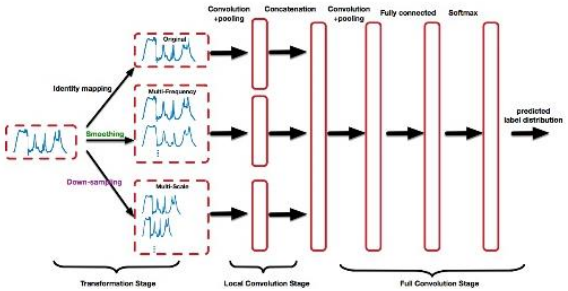
1. Scaling and translation invariance
2. Shift invariance
3. Uniform scaling invariance
4. **Complexity invariance**



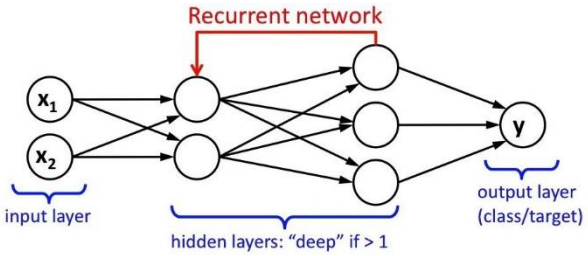
시계열 데이터 분석

- 시계열적 특성을 반영한 일반적인 분석방법은 다음과 같음

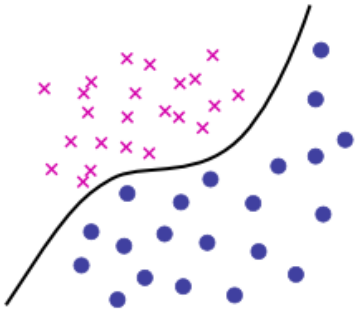
Deep learning 기반 접근법



Convolution Neural Network

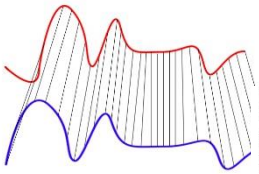


Recurrent Neural Network



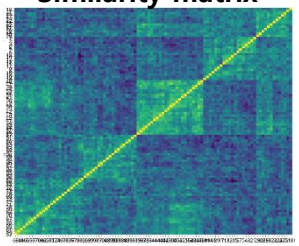
Classification

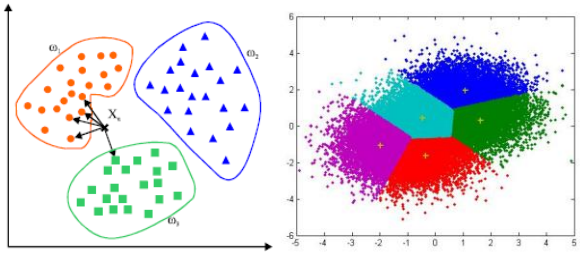
Similarity 기반 접근법



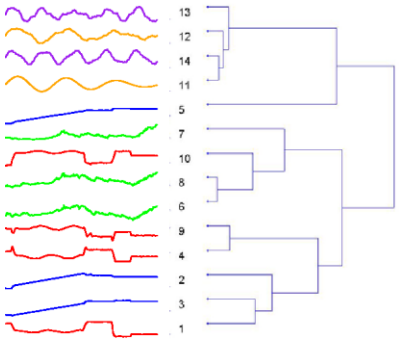
Dynamic Time Warping Matching

Similarity matrix





Distance based methods
(ex, k-means / kNN)



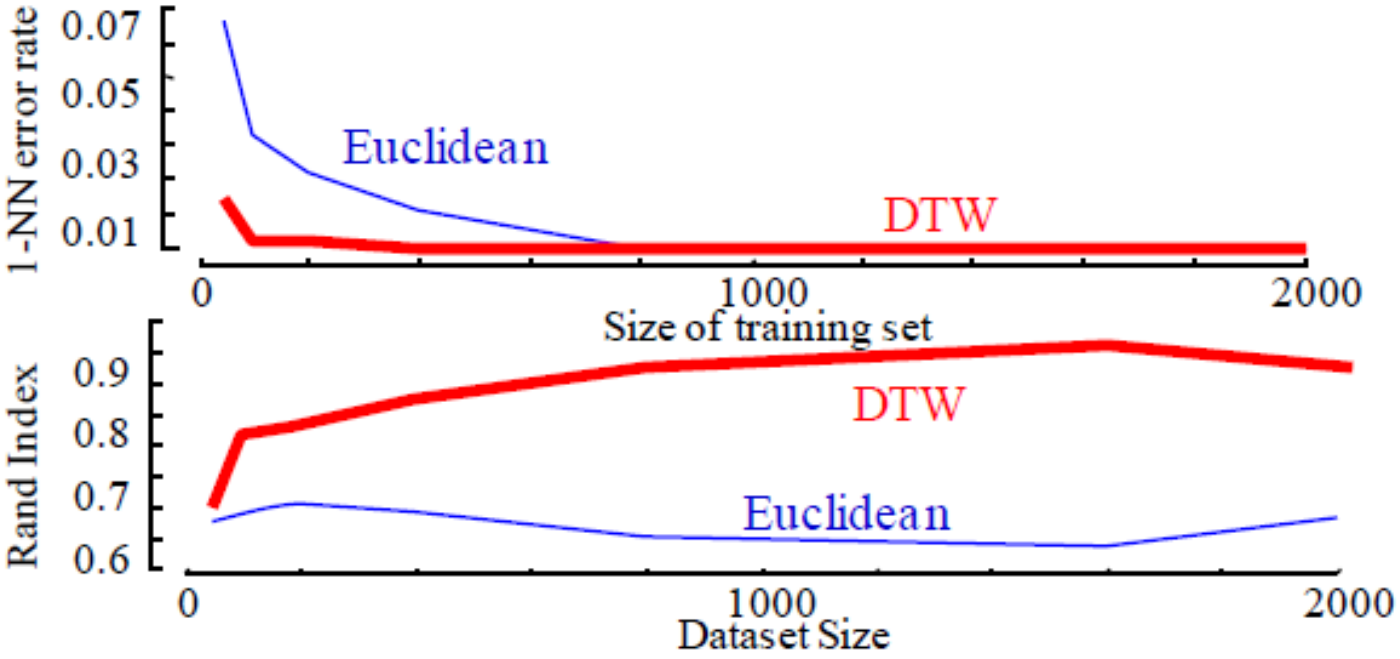
Clustering

시계열 데이터 분석

왜?

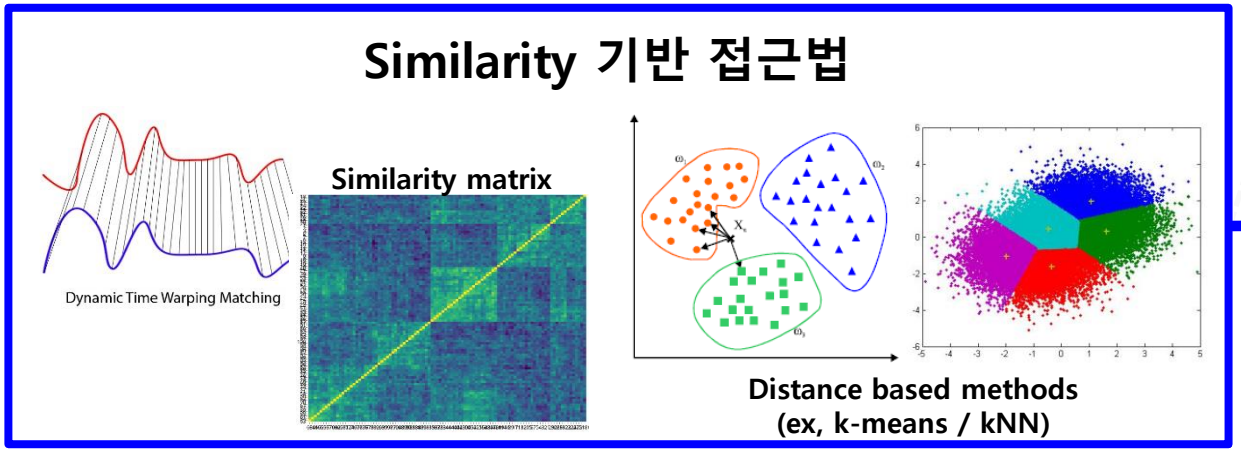
- 딥러닝을 사용한 Classification 모델을 다수가 제안됨
- 반면, Clustering은 최근 연구가 진행되고 있으며, similarity 기반의 접근법이 주로 사용

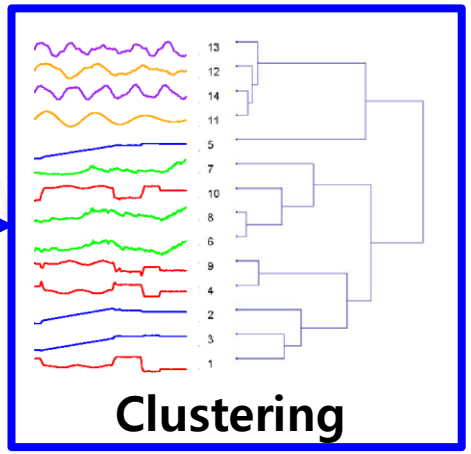
- 시계열간 유사도를 측정하기 위해 DTW(Dynamic time warping)이 사용됨



시계열 데이터 분석

- 시계열적 특성을 반영한 일반적인 분석방법은 다음과 같음

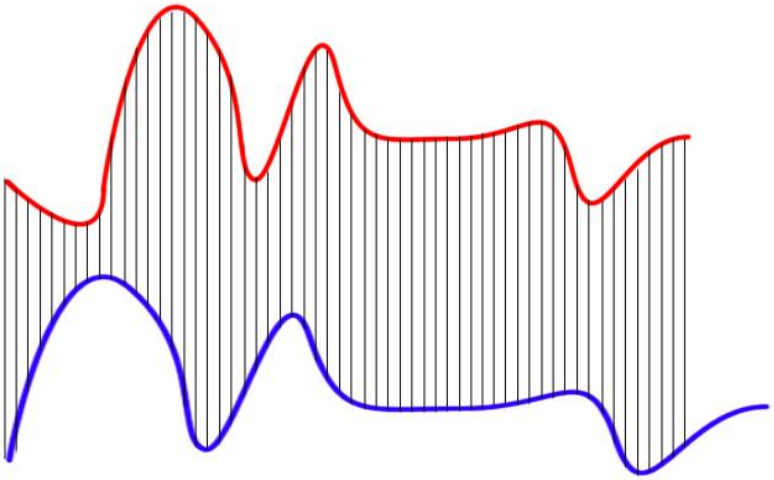




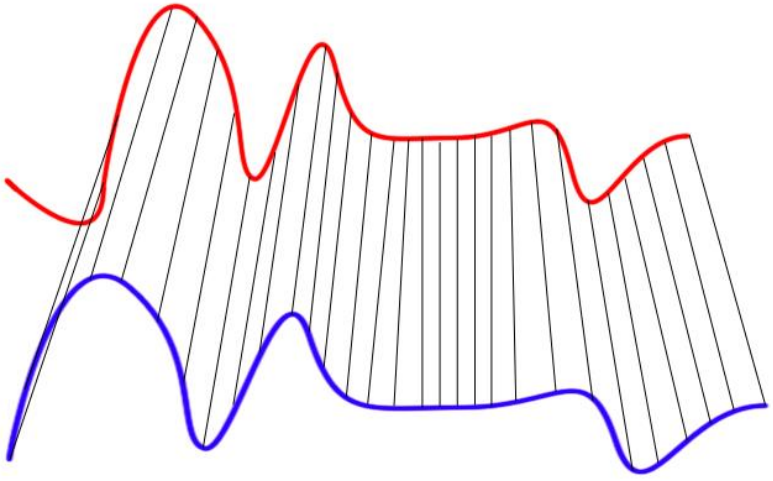
목차

Dynamic time warping

- DTW는 misalignment와 time series length를 고려하여 유사도 산출이 가능



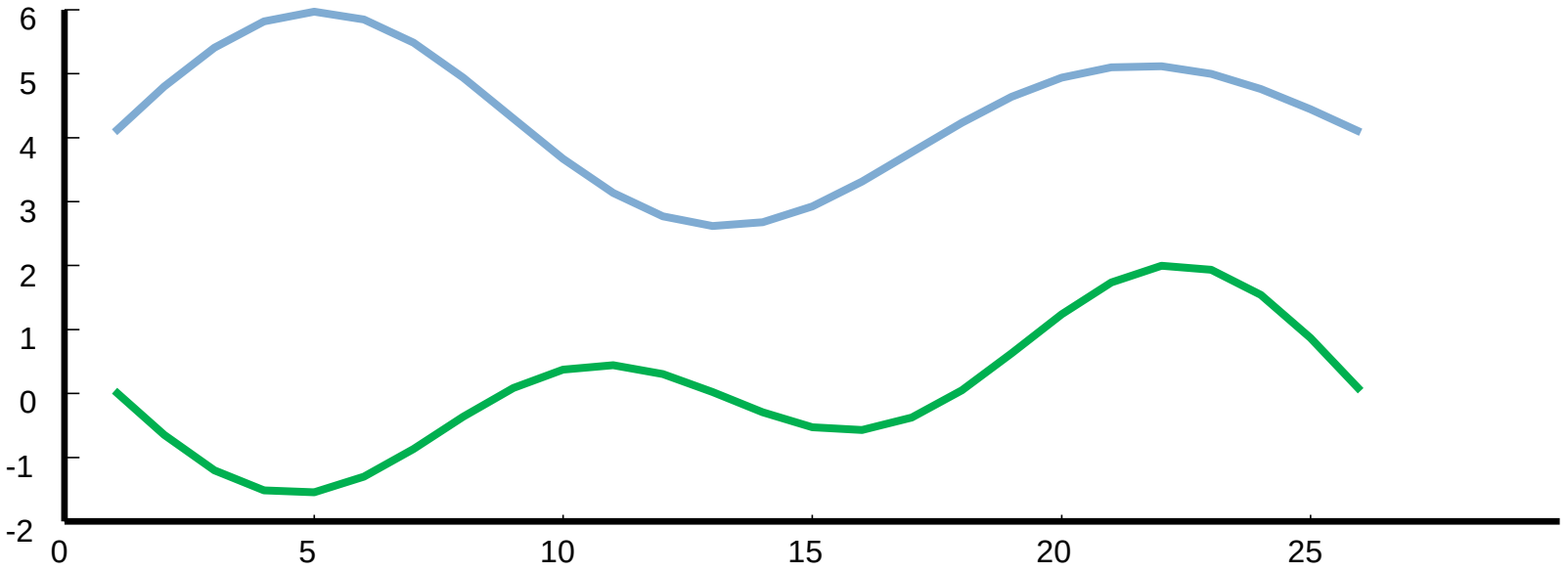
Euclidean Matching



Dynamic Time Warping Matching

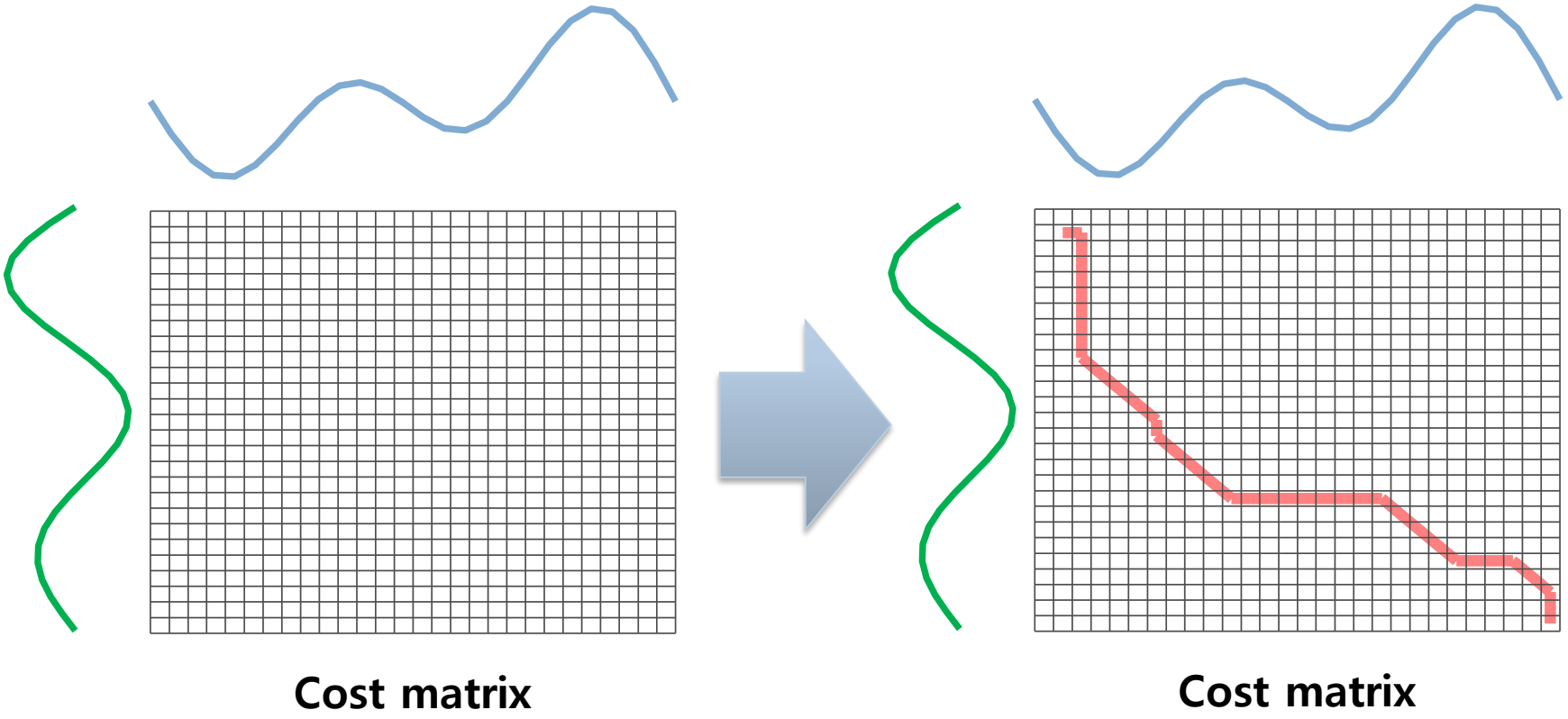
Dynamic time warping

- DTW는 misalignment와 time series length를 고려하여 유사도 산출이 가능



Dynamic time warping

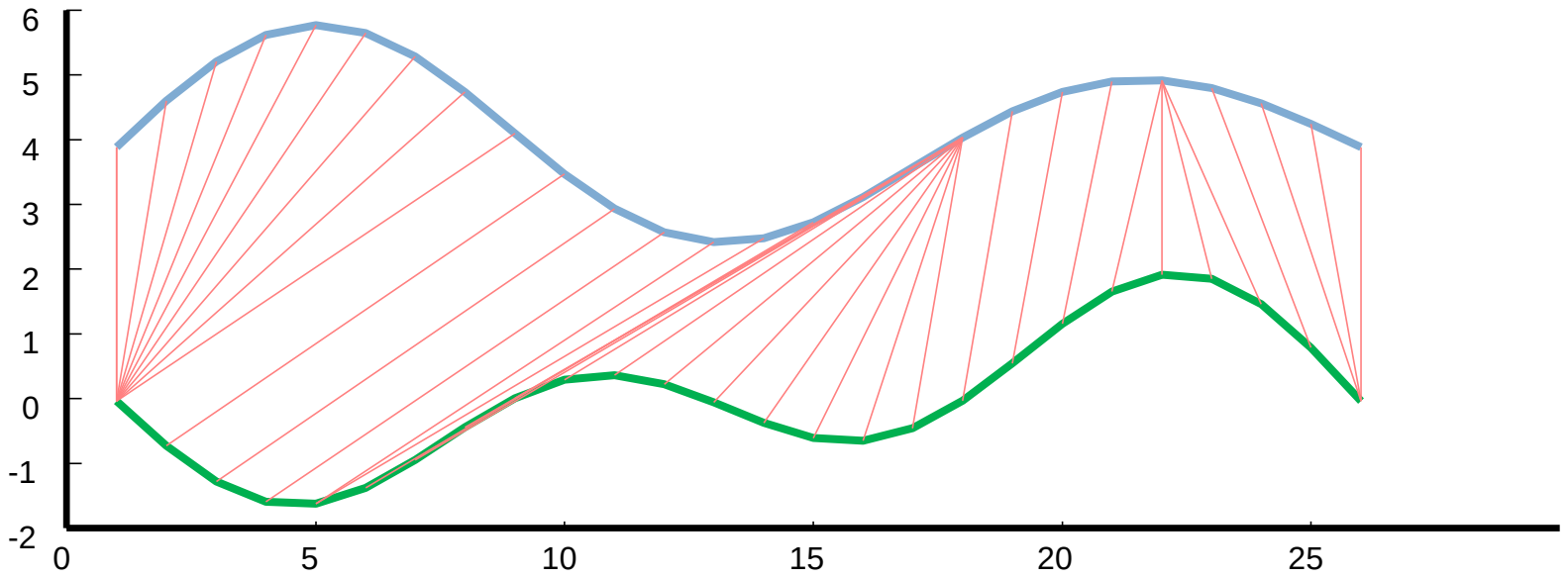
- DTW는 misalignment와 time series length를 고려하여 유사도 산출이 가능
- Cost matrix를 산출하고 총 합이 최소화가 되는 path를 탐색



Cost matrix **Cost matrix**

Dynamic time warping

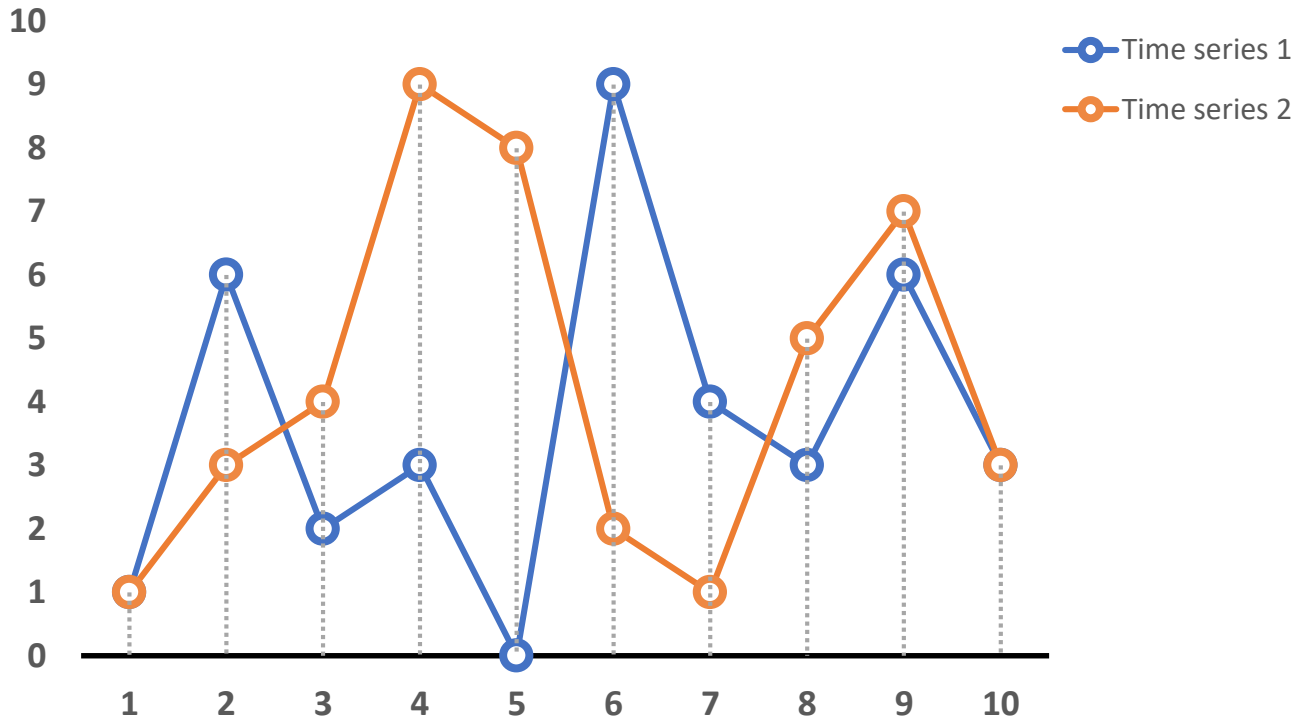
- DTW는 misalignment와 time series length를 고려하여 유사도 산출이 가능



Dynamic time warping

Example

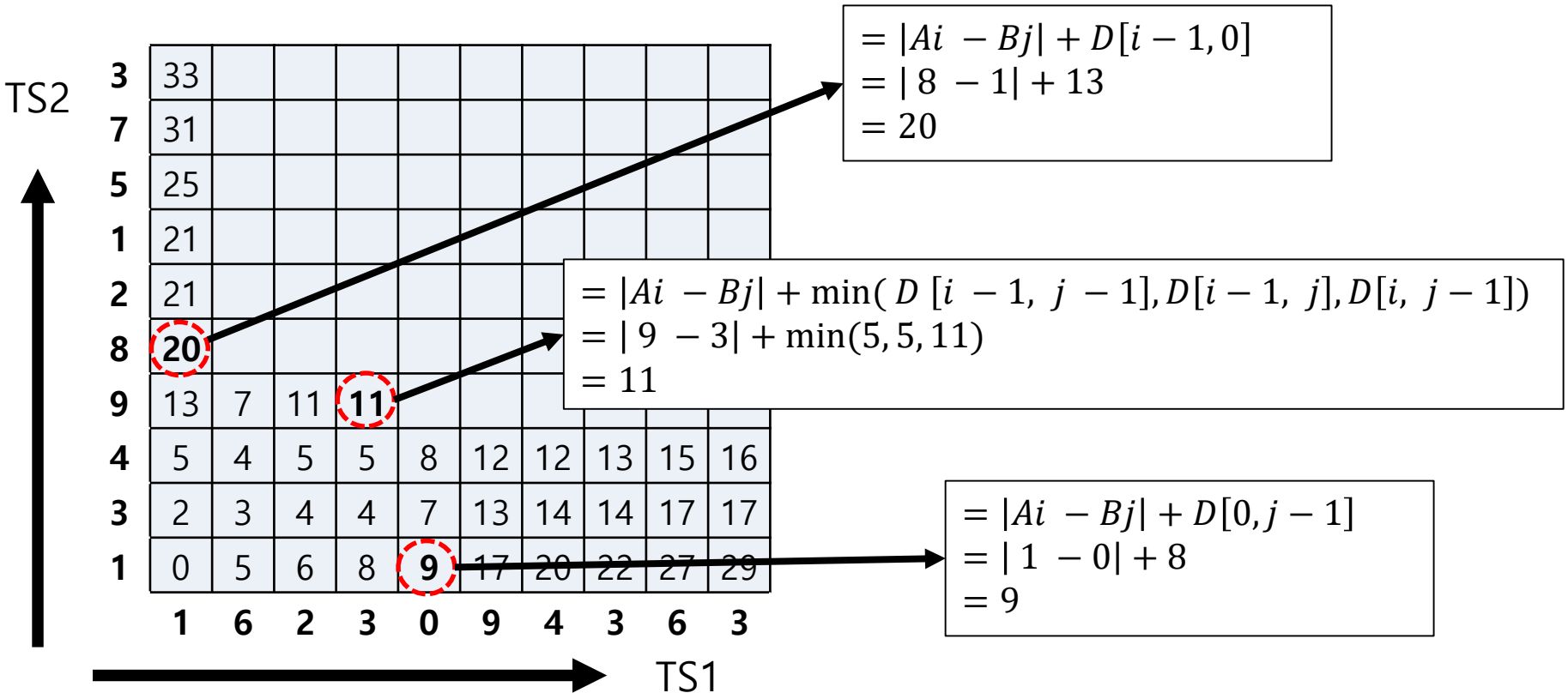
- 2개의 time series 간의 유사도를 산출해보자



Dynamic time warping

Example

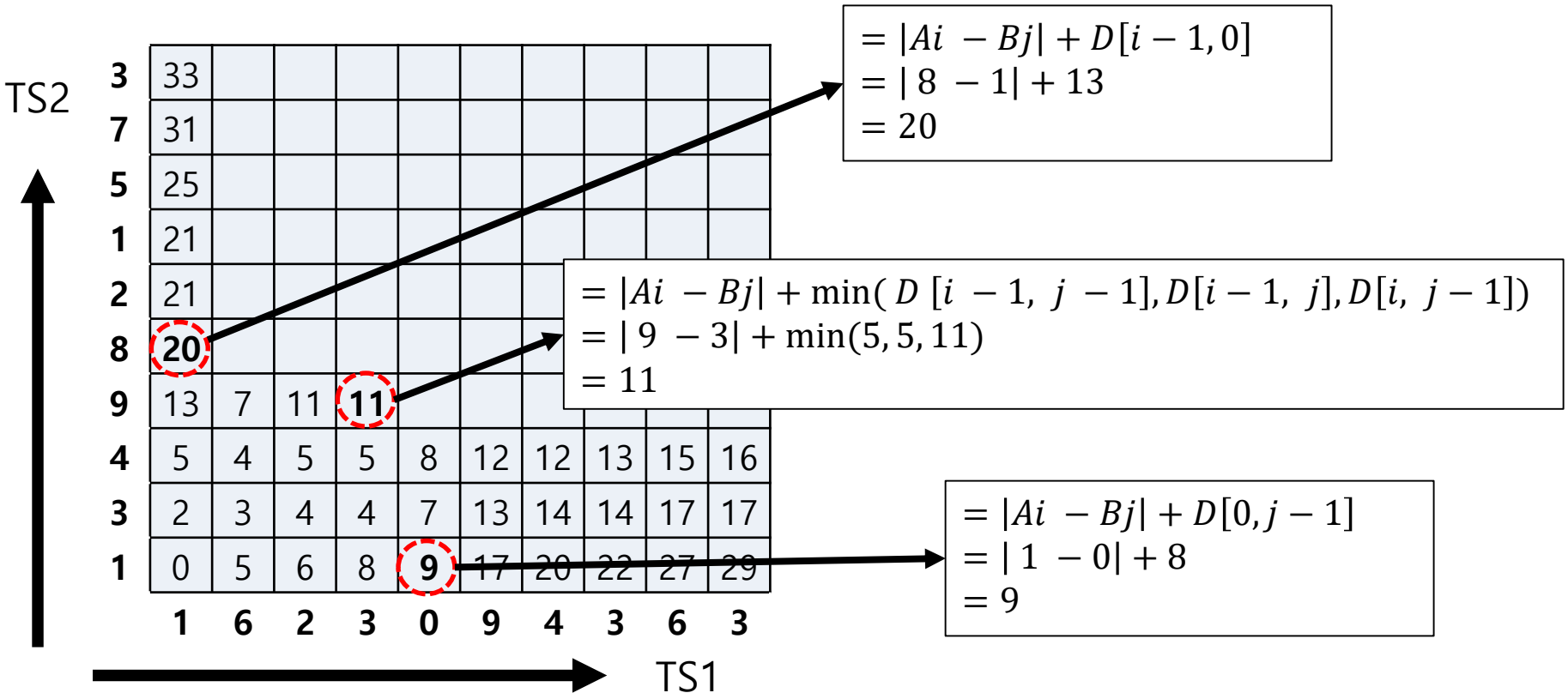
- 2개의 time series 간의 유사도를 산출해보자



Dynamic time warping

Example

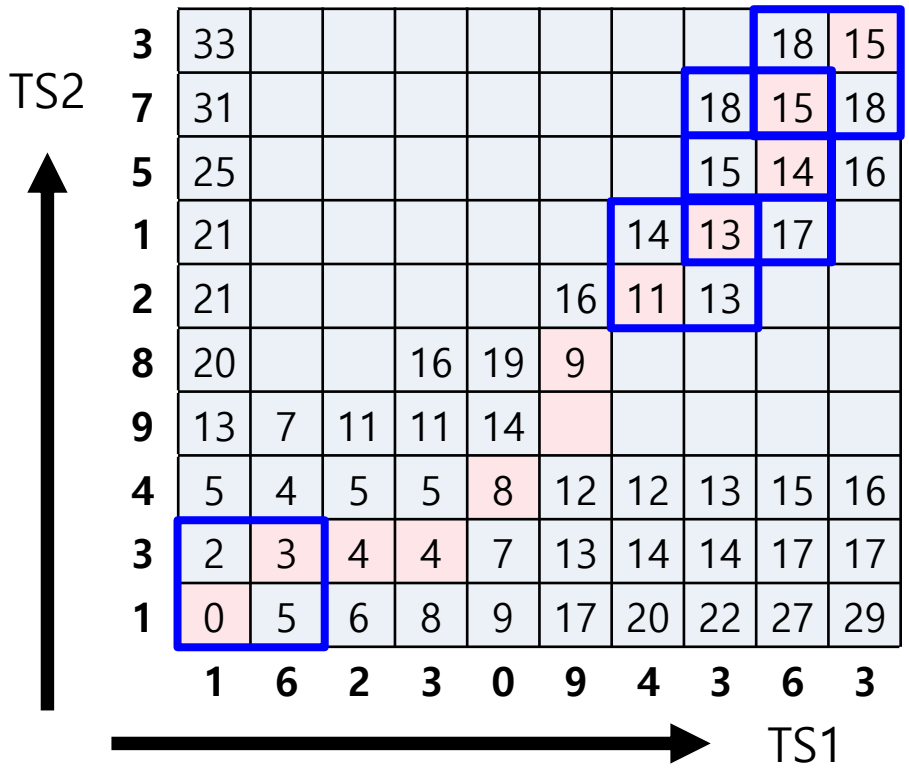
- 2개의 time series 간의 유사도를 산출해보자



Dynamic time warping

Example

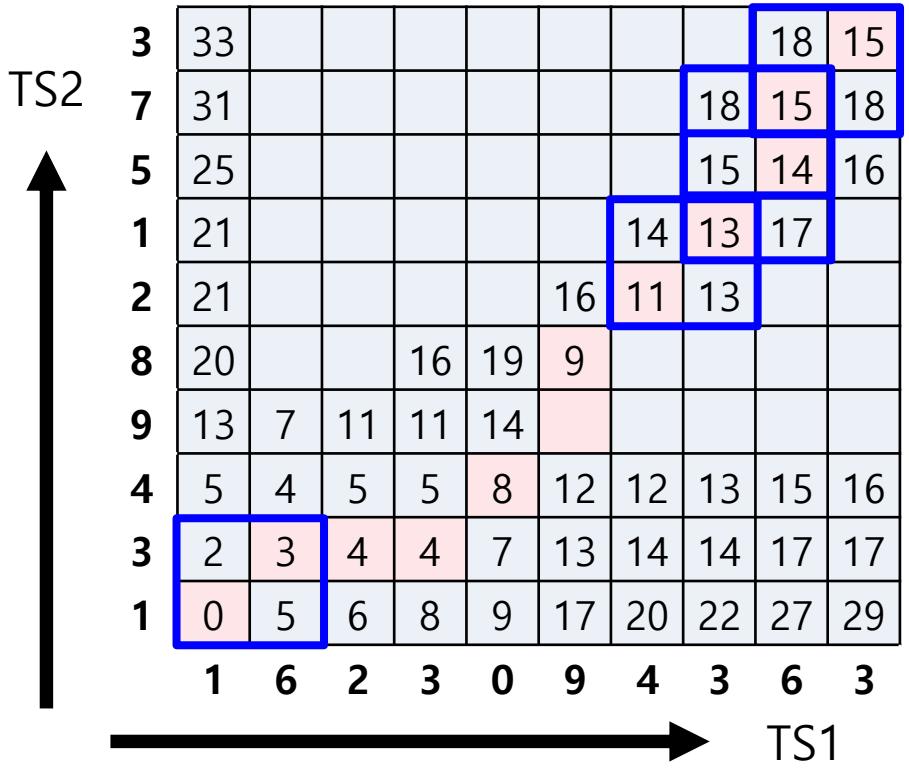
- 끝나는 step까지 총합 (유사도)이 최소가 되도록 path를 설정



Dynamic time warping

drawback

- Time step과 관측치의 개수에 따라 계산 시간이 기하급수적으로 증가



$O(n^2)$ time

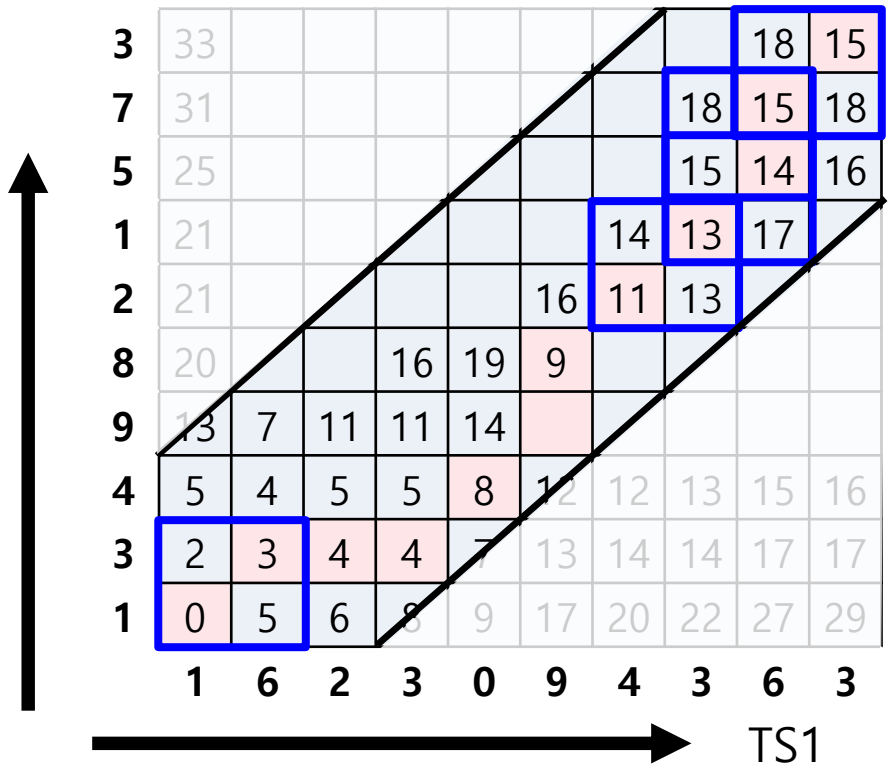
$O(n^2)$ space

목차

Variance of DTW

#1 cDTW (Sakeo-Chiba Band)

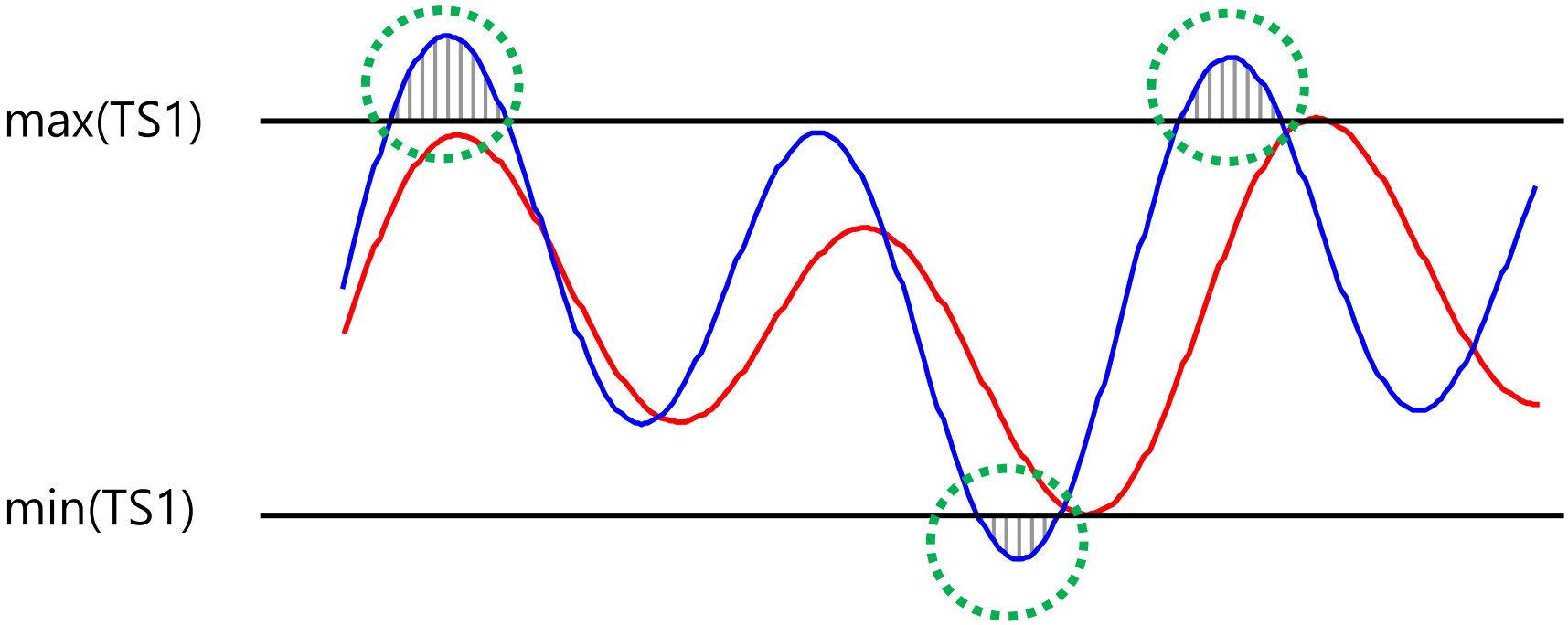
- DTW의 연산을 효율적으로 할 수 있는 다양한 방법이 제안됨



Variance of DTW

#2 Lower bound

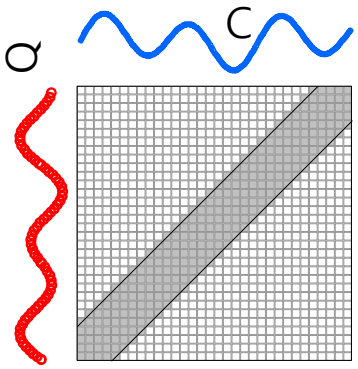
- DTW 계산을 피하고 유사한 효과를 얻기 위해 Boundary 개념을 도입
- Boundary를 넘어가는 부분에 대해서는 ED로 계산하여 계산량을 줄임



Variance of DTW

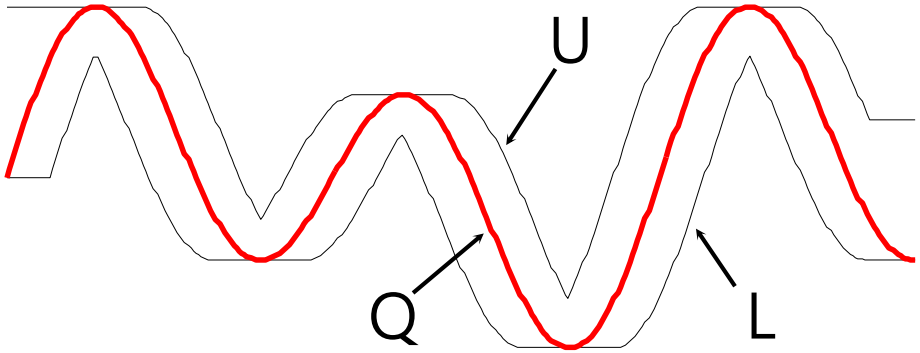
#2 Lower bound of Keogh

- DTW 계산을 피하고 유사한 효과를 얻기 위해 Boundary 개념을 도입
- Boundary를 넘어가는 부분에 대해서는 ED로 계산하여 계산량을 줄임

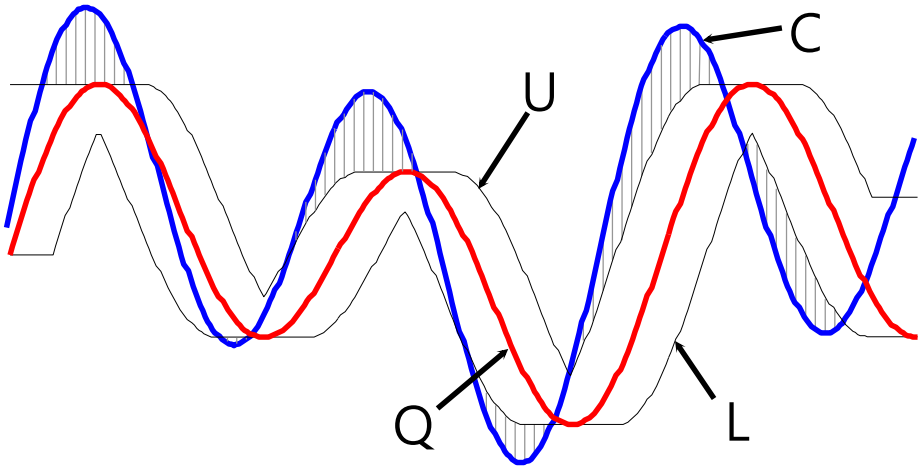


$U_i = \max(q_{i-r} : q_{i+r})$

$L_i = \min(q_{i-r} : q_{i+r})$



$$LB\_Keogh(Q,C) = \sum_{i=1}^{n} \begin{cases} (q_i - U_i)^2 & if\ q_i > U_i \\ (q_i - L_i)^2 & if\ q_i < L_i \\ 0 & otherwise \end{cases}$$



목차

Density Peak

Clustering by fast search and find of density peaks

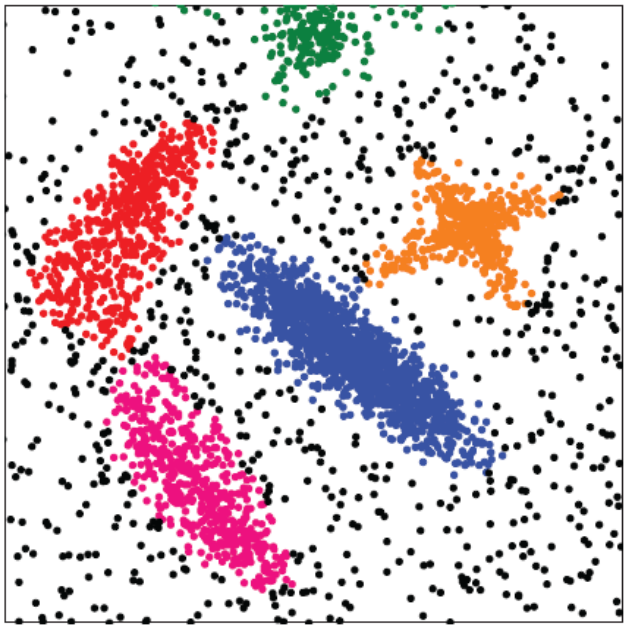
Alex Rodriguez and Alessandro Laio

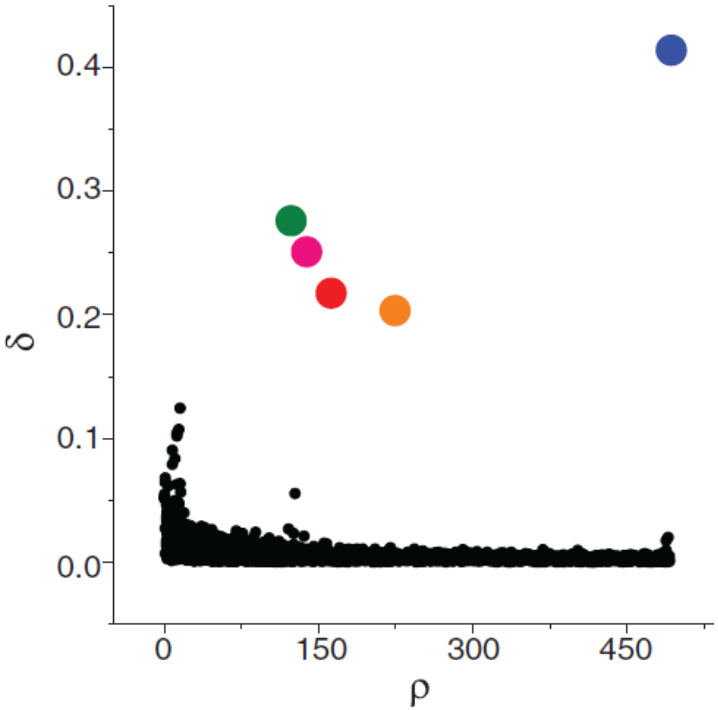
Cluster analysis is aimed at classifying elements into categories on the basis of their similarity. Its applications range from astronomy to bioinformatics, bibliometrics, and pattern recognition. We propose an approach based on the idea that cluster centers are characterized by a higher density than their neighbors and by a relatively large distance from points with higher densities. This idea forms the basis of a clustering procedure in which the number of clusters arises intuitively, outliers are automatically spotted and excluded from the analysis, and clusters are recognized regardless of their shape and of the dimensionality of the space in which they are embedded. We demonstrate the power of the algorithm on several test cases.

Rodriguez, A., & Laio, A. (2014). Clustering by fast search and find of density peaks. *Science*, *344*(6191), 1492-1496.

Density Peak

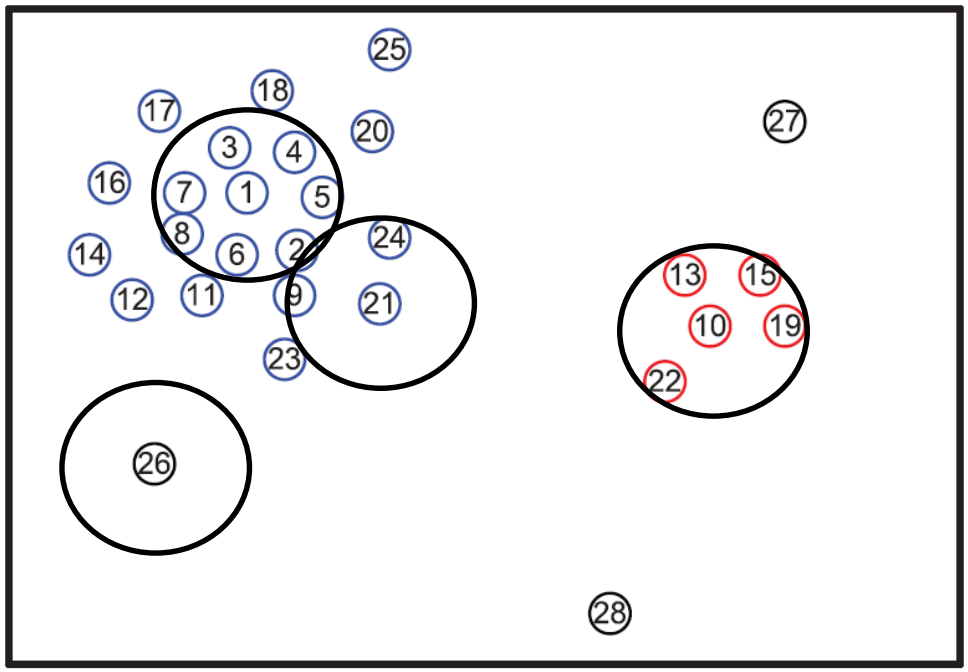
- 밀도 기반의 군집화 방법
- 군집화를 하기 위해 데이터에서 2가지의 요약값을 추출함
  - Local density (ρ_i)
  - Minimum Distance from Points of Higher Density (δ_i)

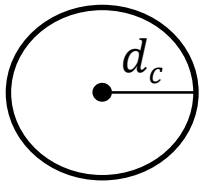




Density Peak

- Local density (ρ_i)
  → 각 관측치 x에 대해 cutoff distance d_c 보다 작은 거리에 있는 관측치 수



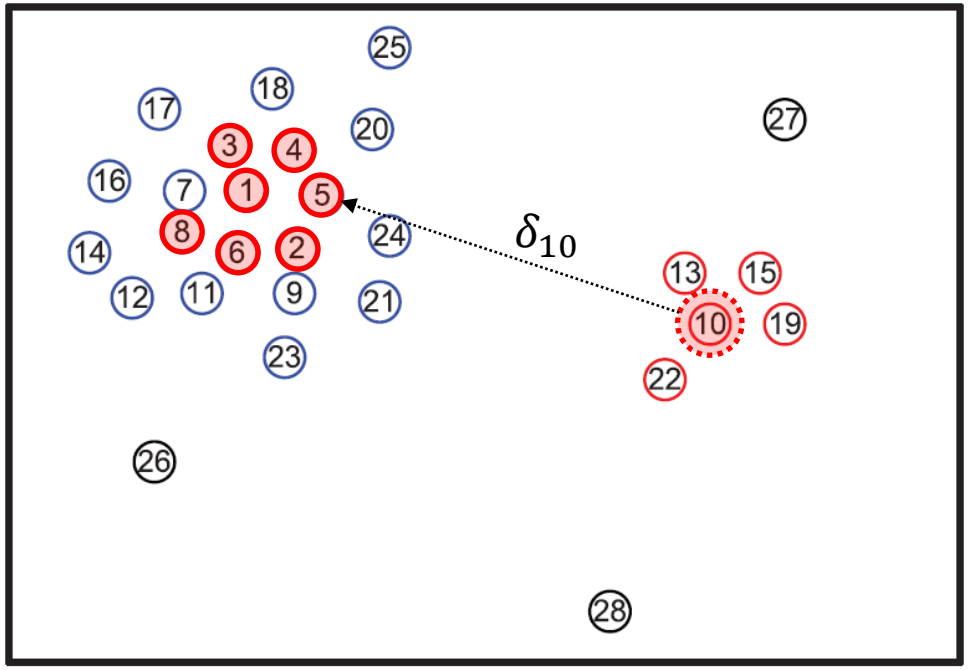


Cutoff distance

Index	ρ_i
1	7
⋮	⋮
10	4
⋮	⋮
21	2
⋮	⋮

Density Peak

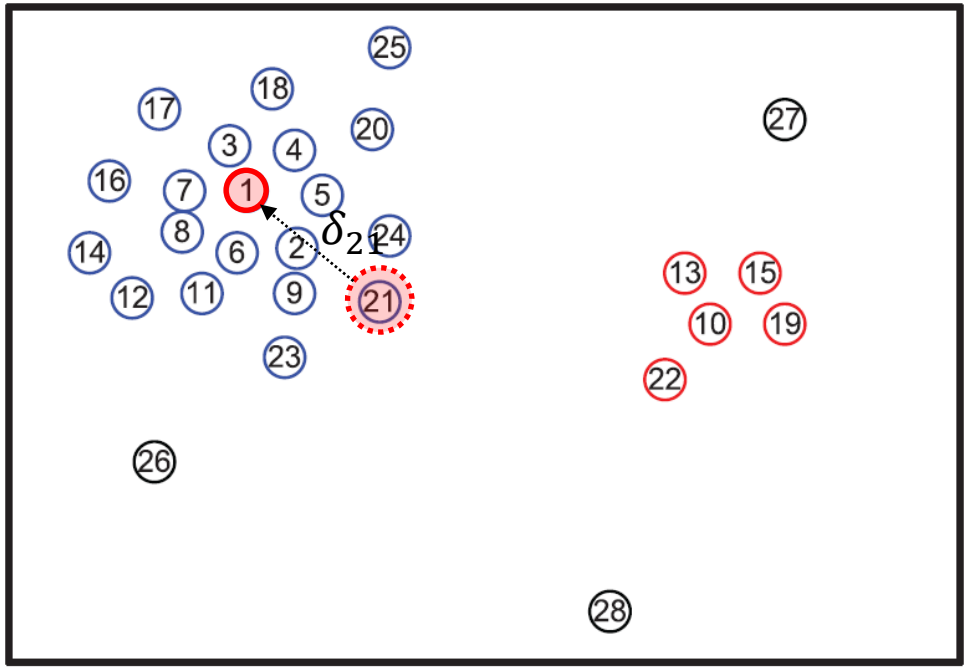
- Minimum Distance from Points of Higher Density (δ_i)
  → 관측치 X의 local density 보다 높은 관측치까지의 거리 중 최소 거리



Index	δ_i
1	1.0
⋮	⋮
10	0.8
⋮	⋮
21	0.18
⋮	⋮

Density Peak

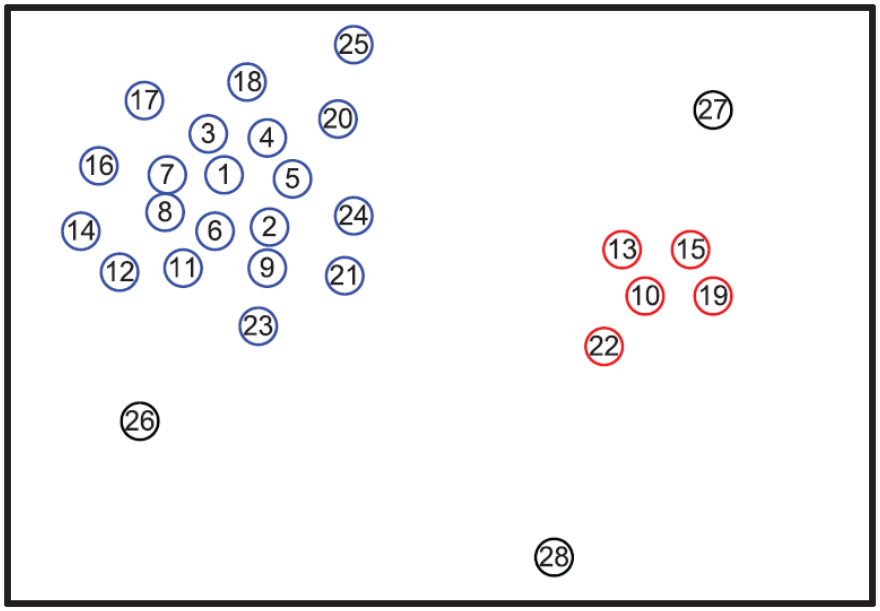
- Minimum Distance from Points of Higher Density (δ_i)
  → 관측치 X의 local density 보다 높은 관측치까지의 거리 중 최소 거리

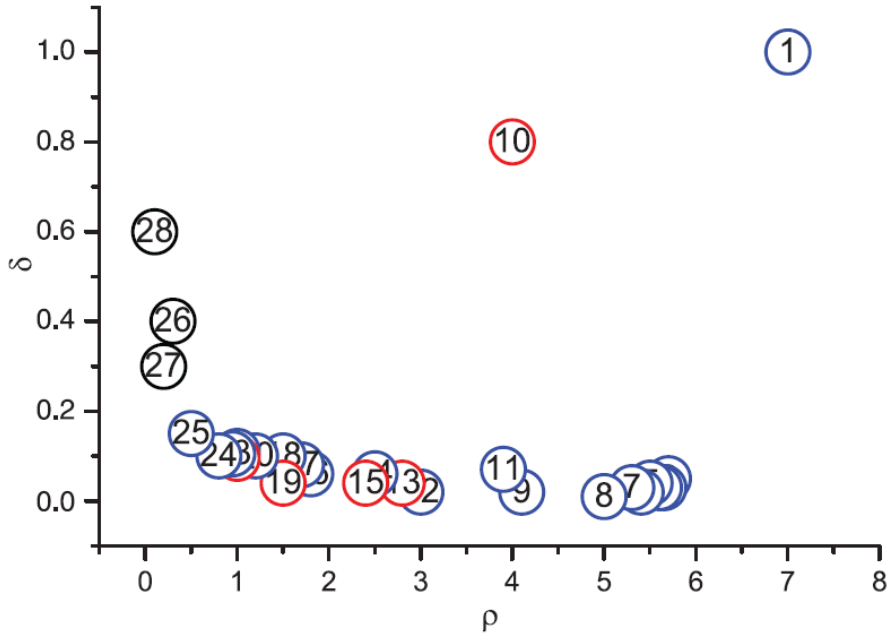


Index	δ_i
1	1.0
⋮	⋮
10	0.8
⋮	⋮
21	0.18
⋮	⋮

Density Peak

- $\delta_i \times \rho_i$에 대해 상위 K개를 군집의 중심으로 선택
  → local density가 높으면서 더 높은 density와는 거리가 멀다
  == 군집의 중심일 수 있다

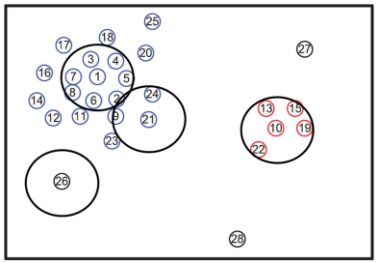




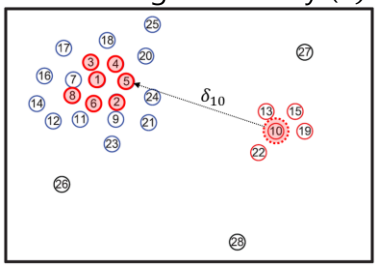
Density Peak

summary

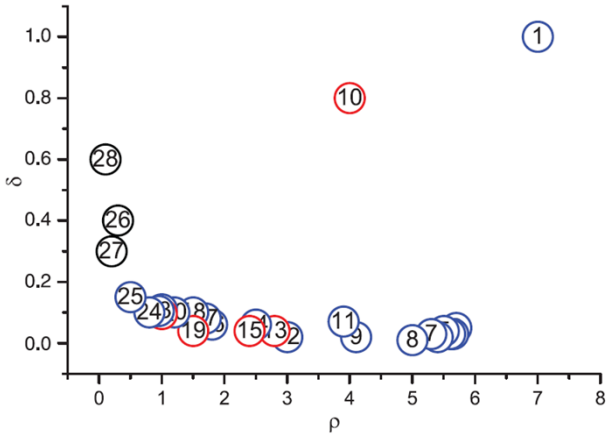
Local density(ρ)



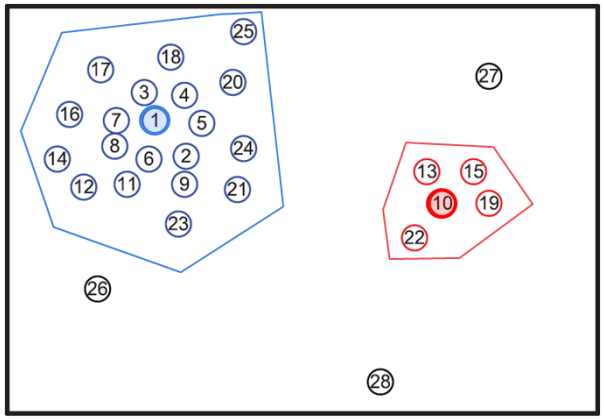
Minimum Distance from
Points of Higher Density (δ)



Decision graph

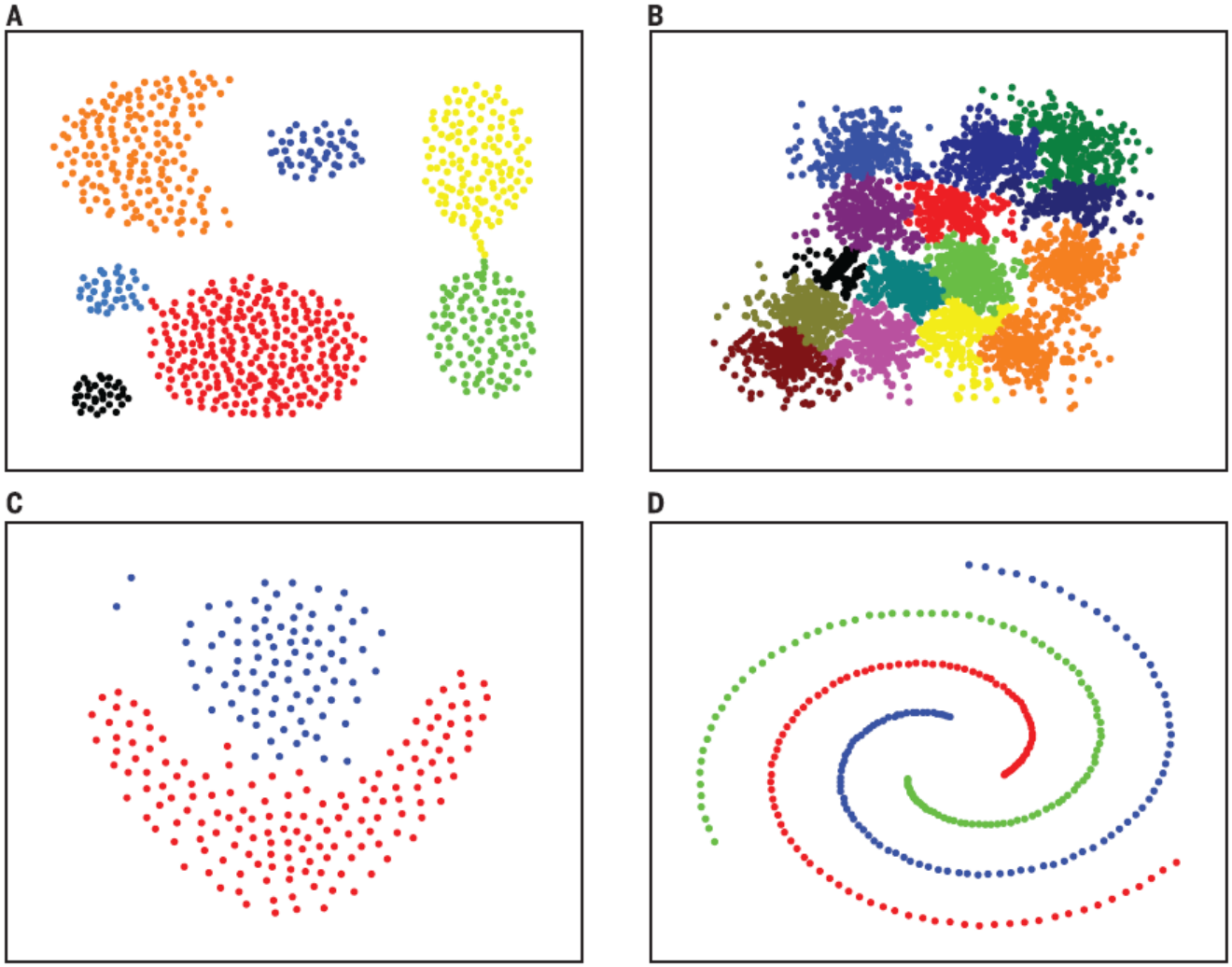


Assign cluster



Density Peak

Experiment results



목차

Tadpole clustering

Accelerating Dynamic Time Warping Clustering with a Novel Admissible Pruning Strategy

Nurjahan Begum Liudmila Ulanova Jun Wang[1] Eamonn Keogh
University of California, Riverside University of Texas at Dallas[1]
{nbegu001, lulan001, eamonn}@cs.ucr.edu wangjun@utdallas.edu[1]

ABSTRACT

Clustering time series is a useful operation in its own right, and an important subroutine in many higher-level data mining analyses, including data editing for classifiers, summarization, and outlier detection. While it has been noted that the general superiority of Dynamic Time Warping (DTW) over Euclidean Distance for *similarity search* diminishes as we consider ever larger datasets, as we shall show, the same is not true for *clustering*. Thus, clustering time series under DTW remains a computationally challenging task. In this work, we address this lethargy in two ways. We propose a novel pruning strategy that exploits both upper and lower bounds to prune off a large fraction of the expensive distance calculations. This pruning strategy is admissible; giving us provably identical results to the brute force algorithm, but is at least an order of magnitude faster. For datasets where even this level of speedup is inadequate, we show that we can use a simple heuristic to order the unavoidable calculations in a *most-useful-first* ordering, thus casting the clustering as an anytime algorithm. We demonstrate the utility of our ideas with both single and multidimensional case studies in the domains of

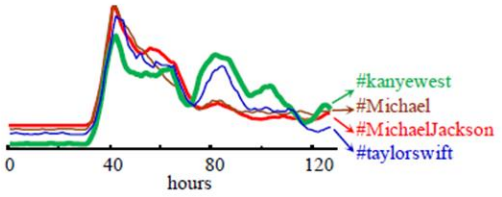


Figure 1: A cluster of four Twitter hashtag usage time series (normalized for volume) over ~6 days starting from June 12, 2009 [32]. (Best viewed in color.)

- **Association Discovery**: Here we see that *#kanyewest* and *#taylorswift* have highly similar time series representations, but are clearly not synonyms. If we test to see whether this relationship existed prior to the illustrated timeframe, we find it does not. This suggests the existence of an *event* that caused this temporary association, and with a little work we can discover the famous "*I'mma let you finish*" event at the 2009 Video Music Awards [35].

Begum, N., Ulanova, L., Wang, J., & Keogh, E. (2015, August). Accelerating dynamic time warping clustering with a novel admissible pruning strategy. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 49-58). ACM.

Tadpole clustering

summary

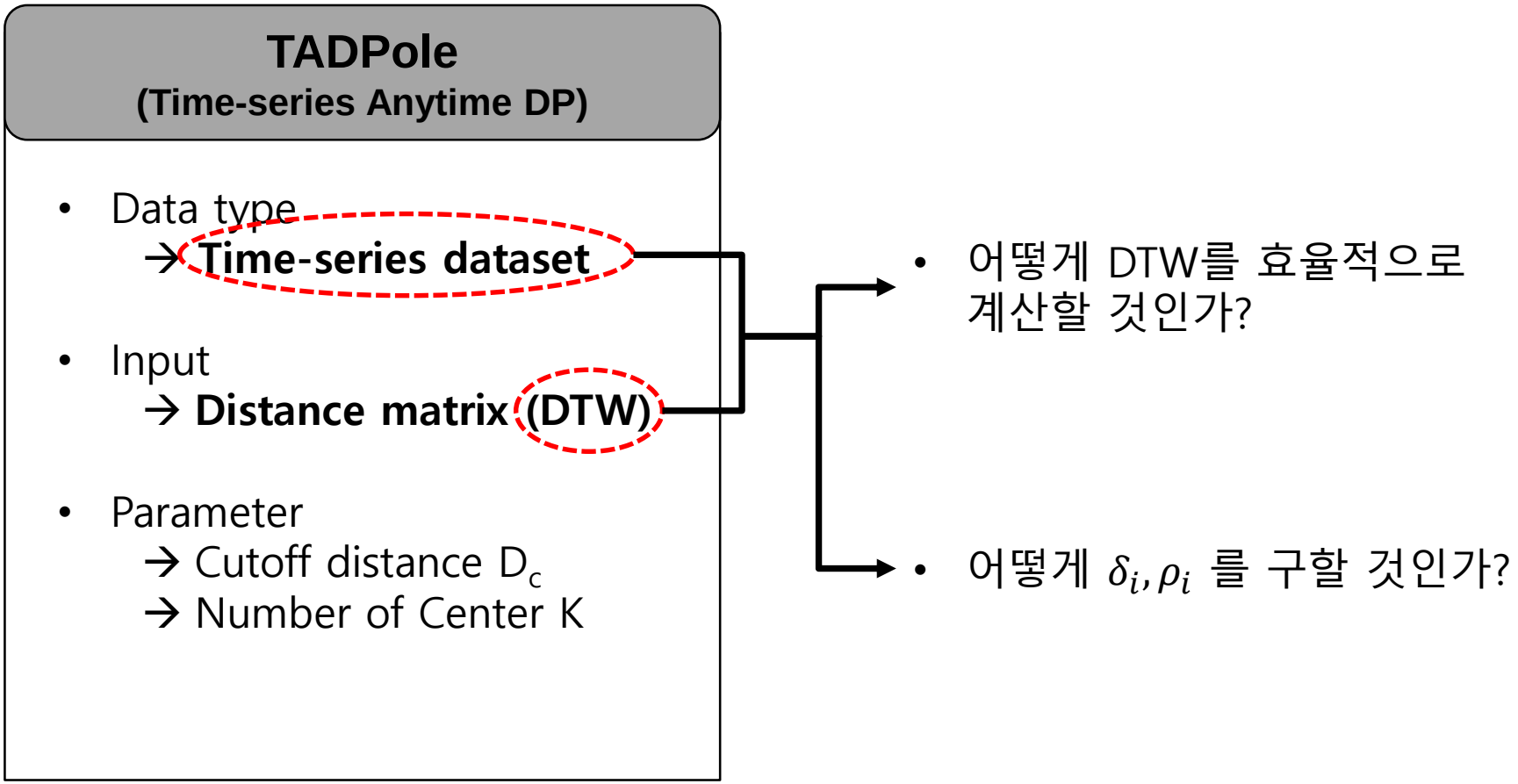
- Time series 데이터 적용가능한 Density peak 알고리즘

Density Peak	TADPole (Time-series Anytime DP)
• Data type → **Multivariate dataset**	• Data type → **Time-series dataset**
• Input → **Distance matrix (ED)**	• Input → **Distance matrix (DTW)**
• Parameter → Cutoff distance D_c → Number of Center K	• Parameter → Cutoff distance D_c → Number of Center K

Tadpole clustering

object

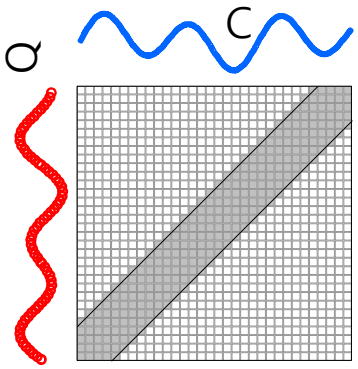
- Time series 데이터 적용가능한 Density peak 알고리즘



(remind) Variance of DTW

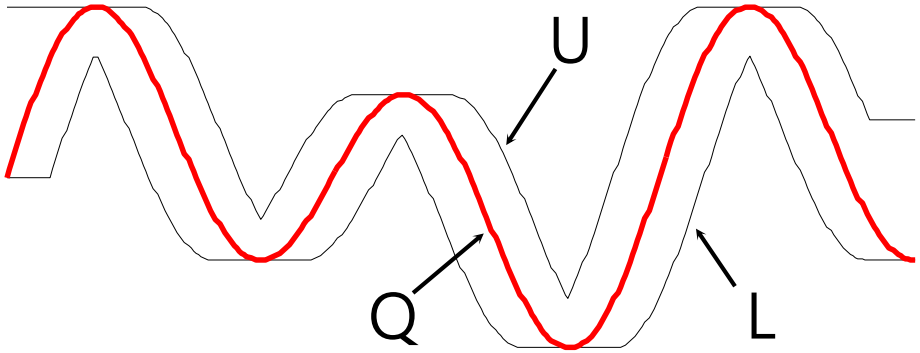
#2 Lower bound of Keogh

- DTW 계산을 피하고 유사한 효과를 얻기 위해 Boundary 개념을 도입
- Boundary를 넘어가는 부분에 대해서는 ED로 계산하여 계산량을 줄임

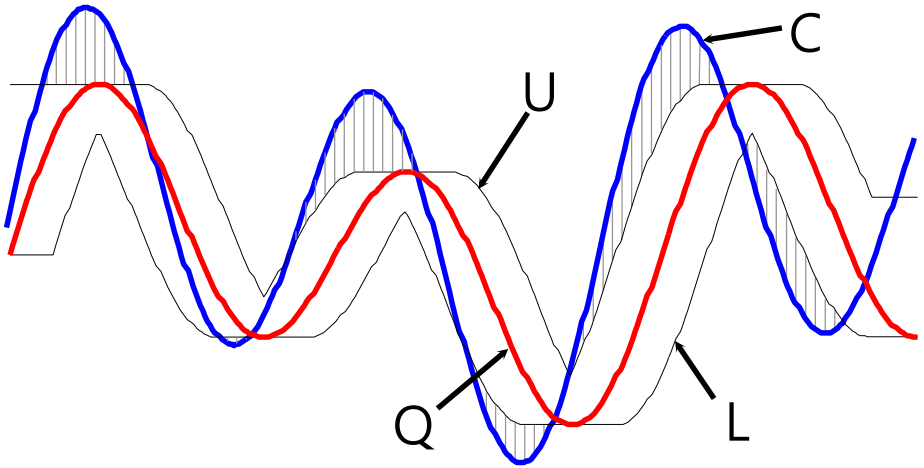


$U_i = \max(q_{i-r} : q_{i+r})$

$L_i = \min(q_{i-r} : q_{i+r})$



$$LB\_Keogh(Q,C) = \sum_{i=1}^{n} \begin{cases} (q_i - U_i)^2 & if\ q_i > U_i \\ (q_i - L_i)^2 & if\ q_i < L_i \\ 0 & otherwise \end{cases}$$



Tadpole clustering

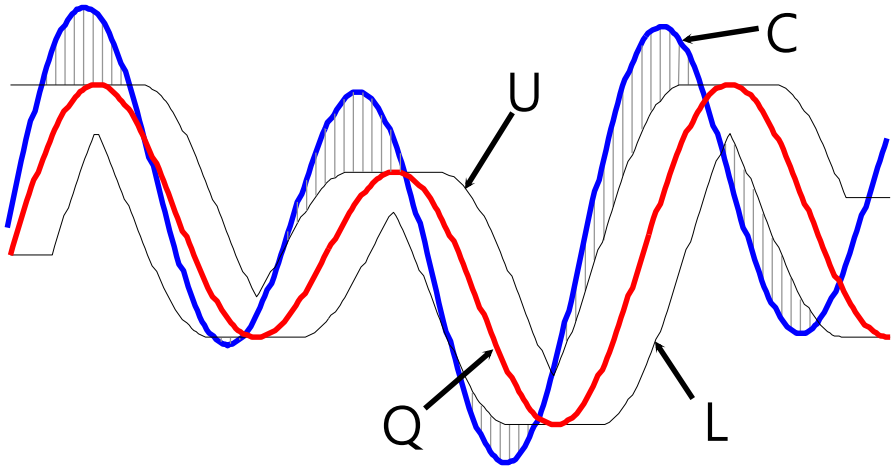
시작에 앞서..

- DTW의 연산을 피하기 위해 다음을 정의함

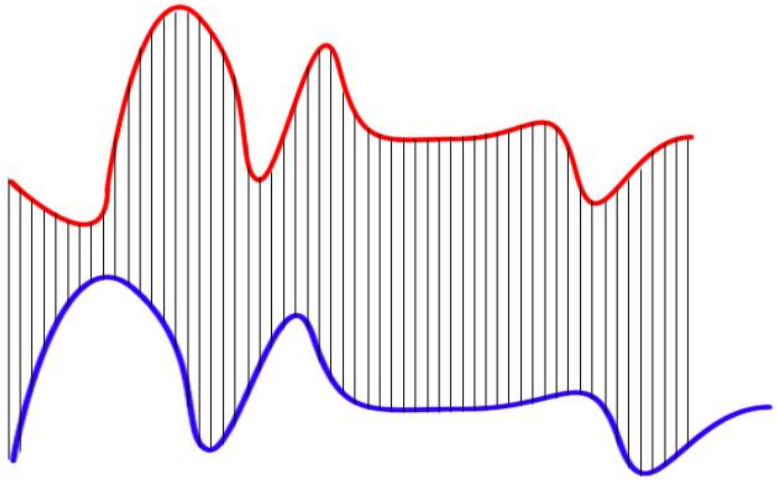
  → Lb_{matrix} = LB_Keogh

  → Ub_{matrix} = Euclidean distance

  ➔ $Lb_{matrix} \leq D_{ij} \leq Ub_{matrix,}$ D_{ij} = DTW distance



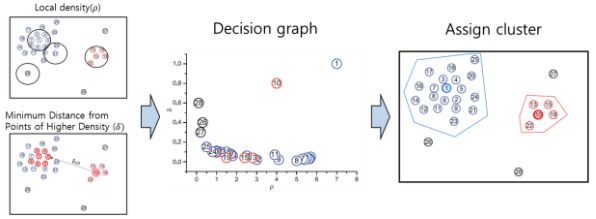
Lb_{matrix}

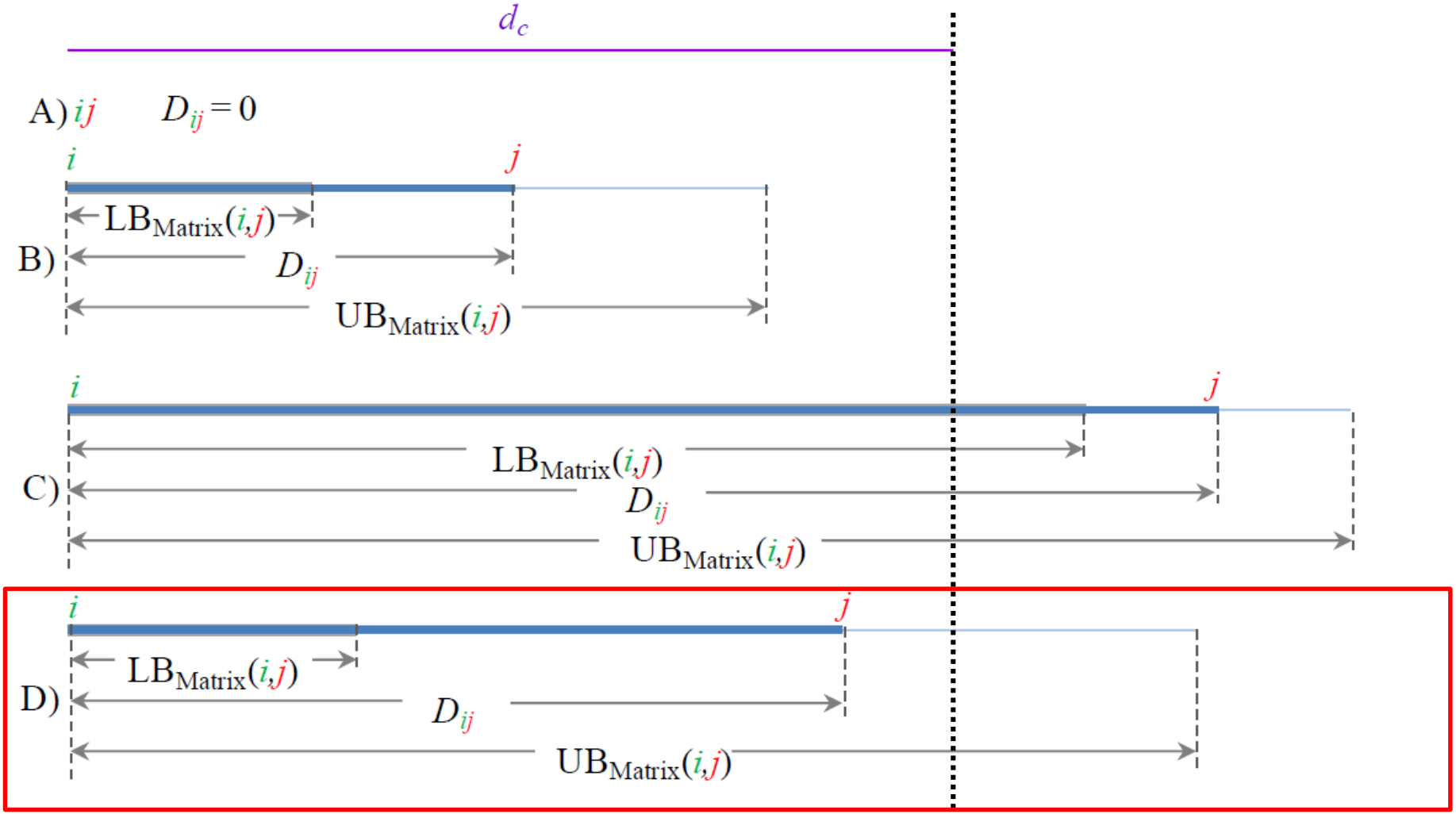


UB_{matrix}

Tadpole clustering

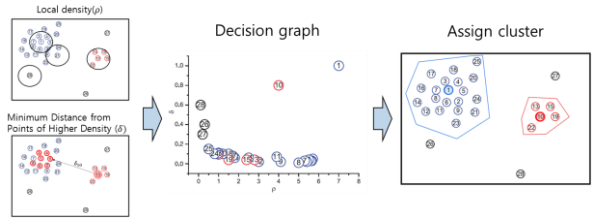
Local density





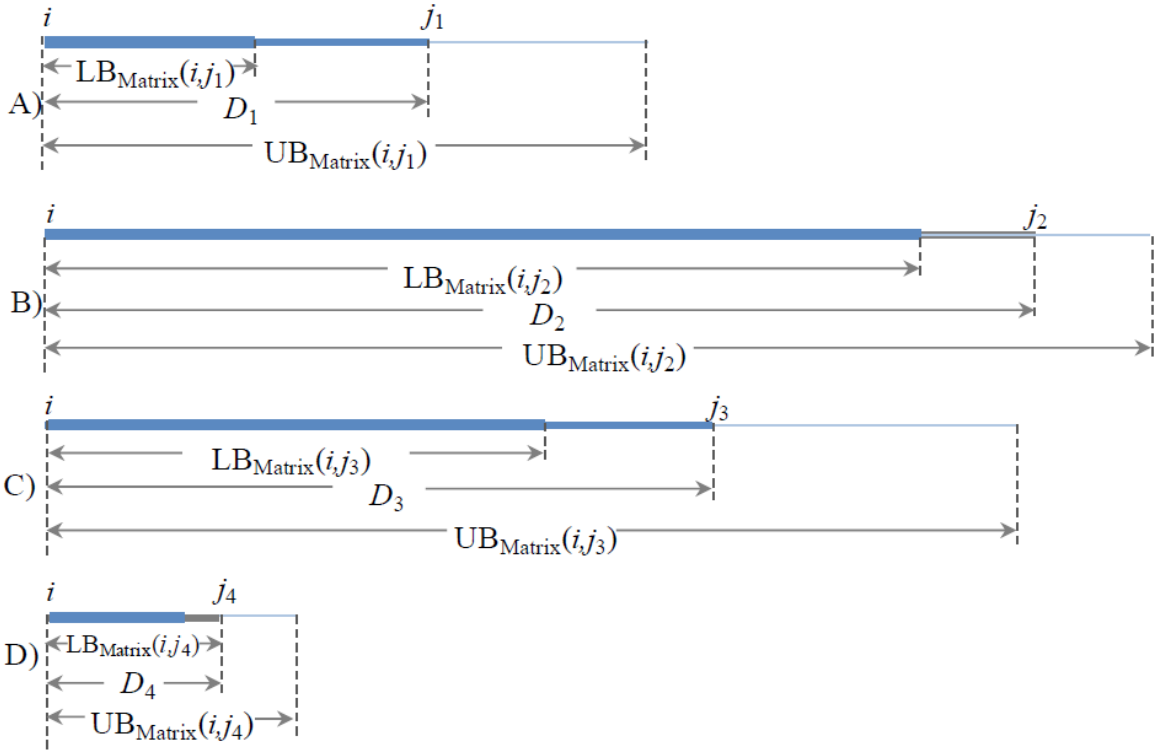
Tadpole clustering

Minimum Distance from Points of Higher Density



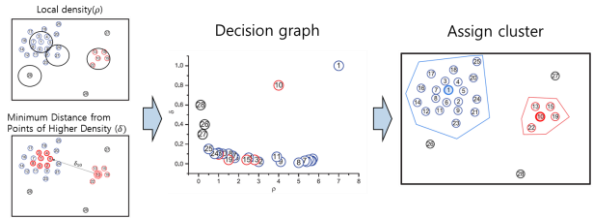
- Minimum Distance from Points of Higher Density (δ_i)
  → 관측치 X의 local density 보다 높은 관측치까지의 거리 중 최소 거리

Initial distance = Inf



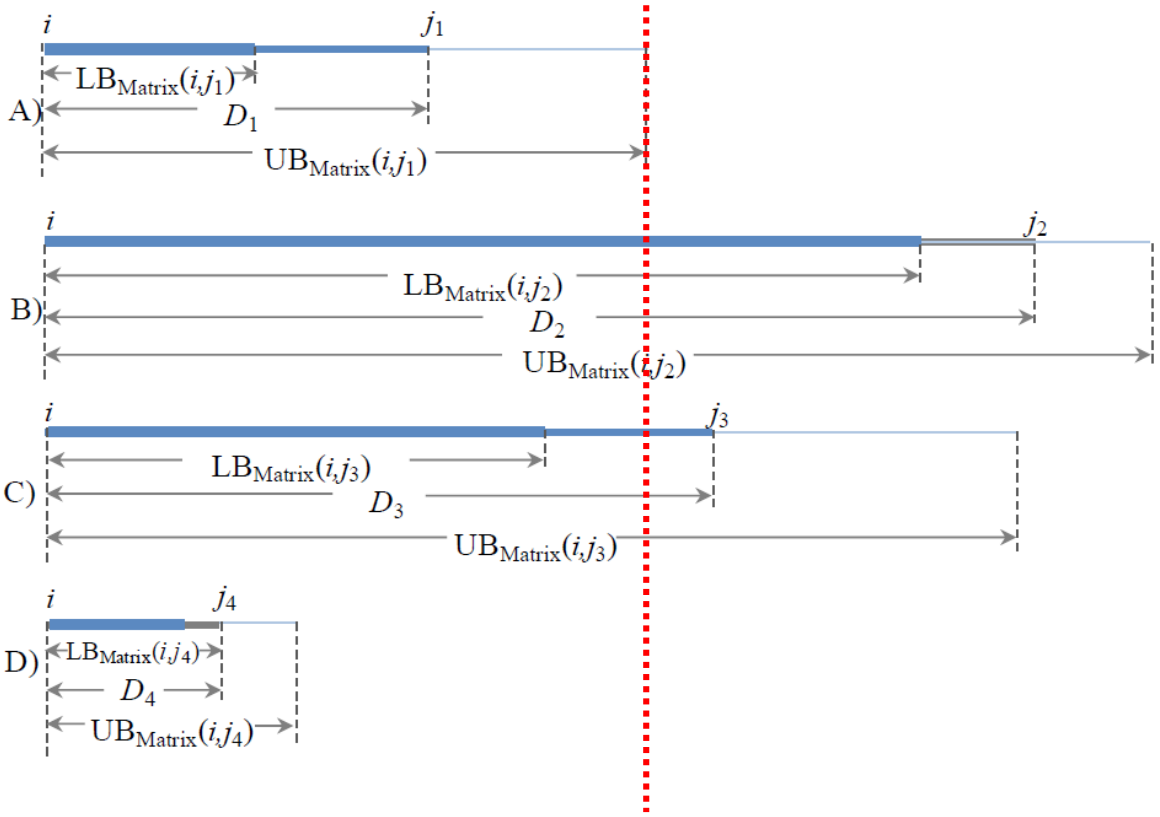
Tadpole clustering

Minimum Distance from Points of Higher Density



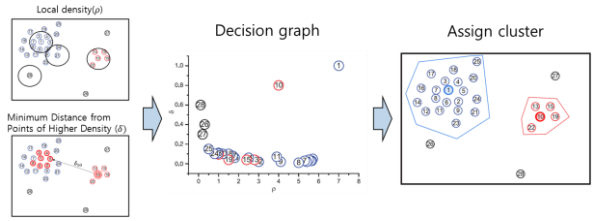
- Minimum Distance from Points of Higher Density (δ_i)
  → 관측치 X의 local density 보다 높은 관측치까지의 거리 중 최소 거리

updated distance = $Ub_{matrix}(i, j_1)$



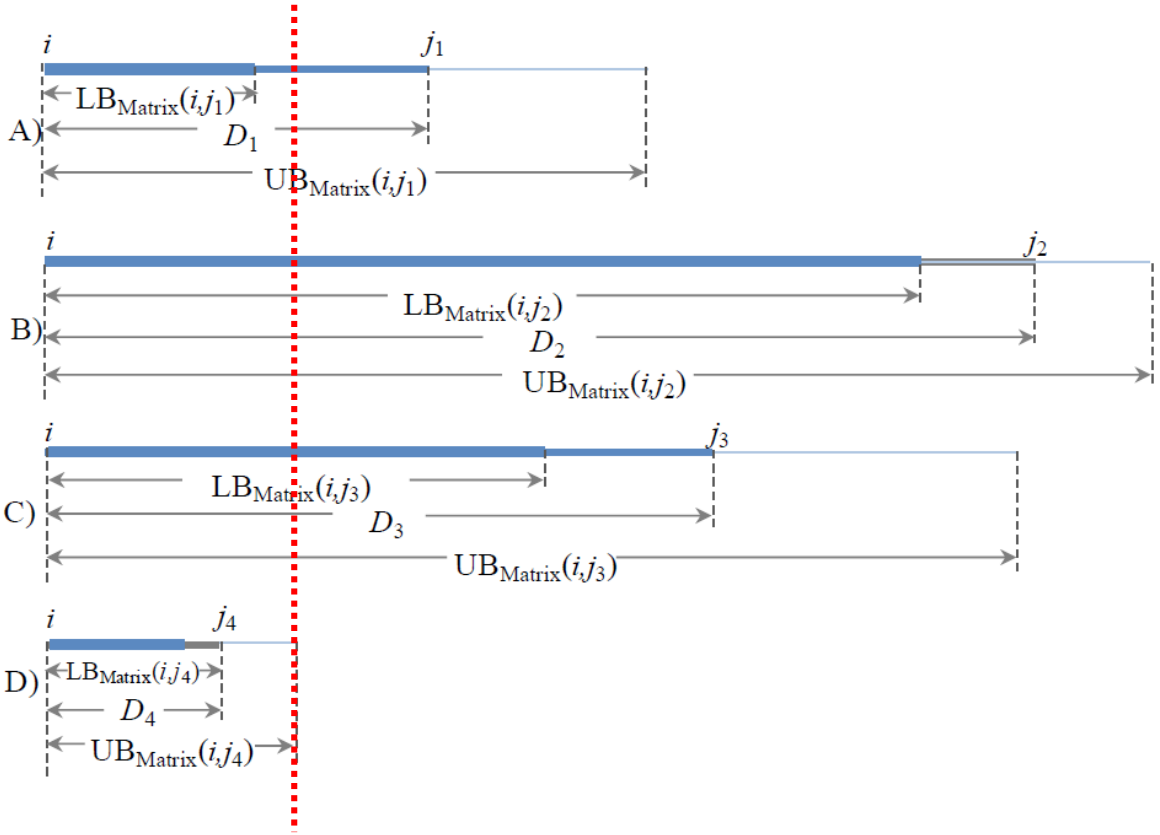
Tadpole clustering

Minimum Distance from Points of Higher Density



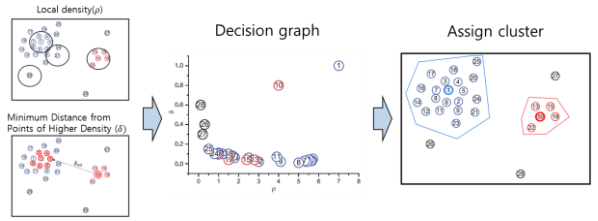
- Minimum Distance from Points of Higher Density (δ_i)
  → 관측치 X의 local density 보다 높은 관측치까지의 거리 중 최소 거리

updated distance = $Ub_{matrix}(i, j_4)$



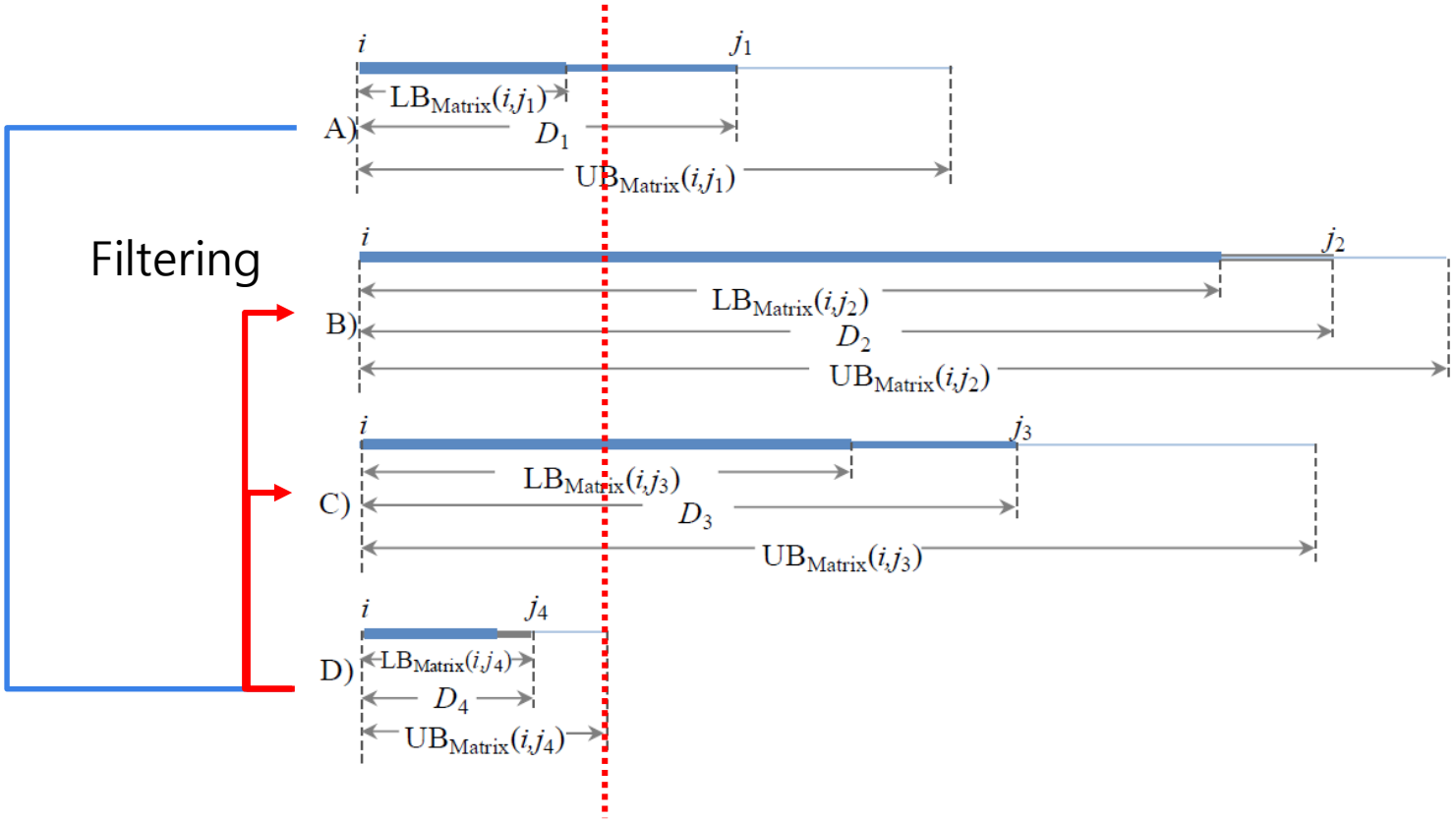
Tadpole clustering

Minimum Distance from Points of Higher Density



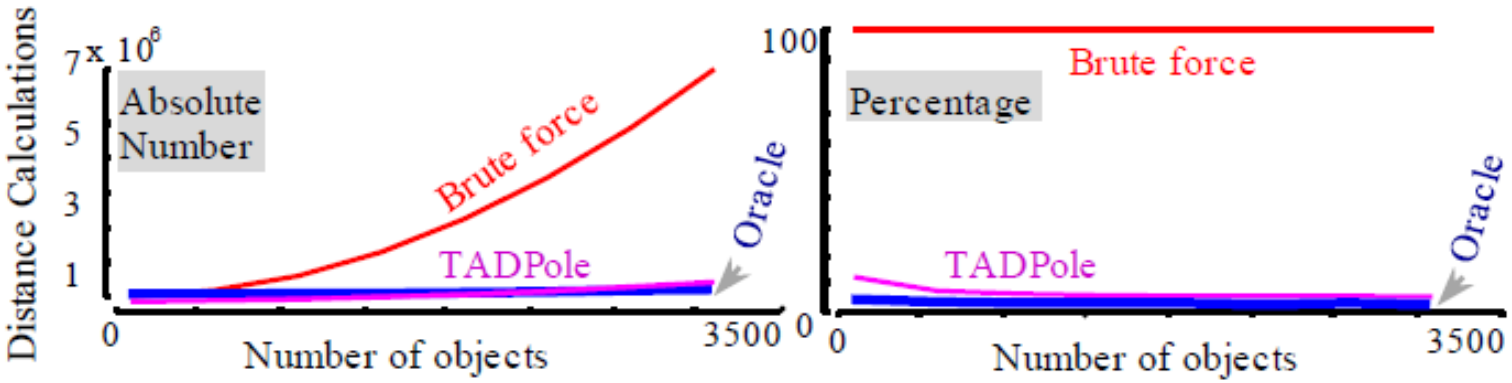
- Minimum Distance from Points of Higher Density (δ_i)
  → 관측치 X의 local density 보다 높은 관측치까지의 거리 중 최소 거리

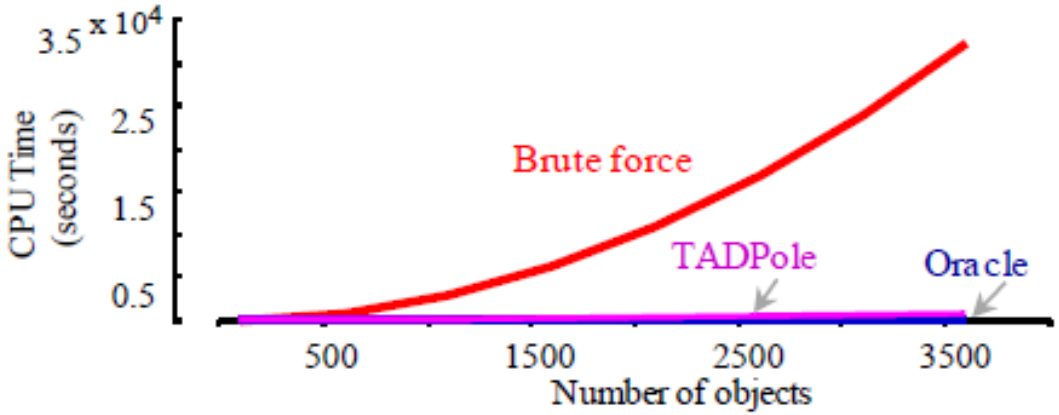
updated distance = $Ub_{matrix}(i, j_4)$



Tadpole clustering

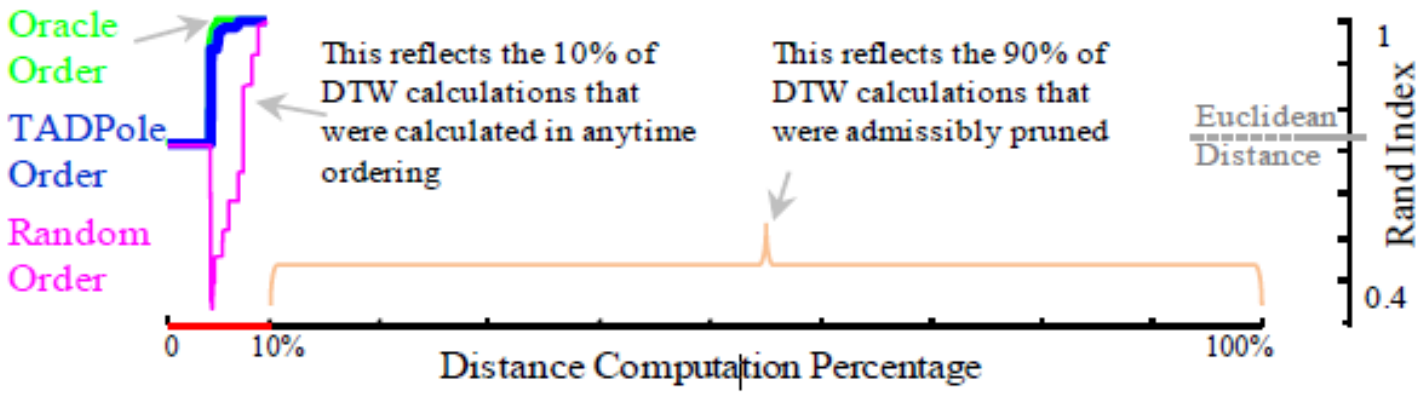
Summary & Results

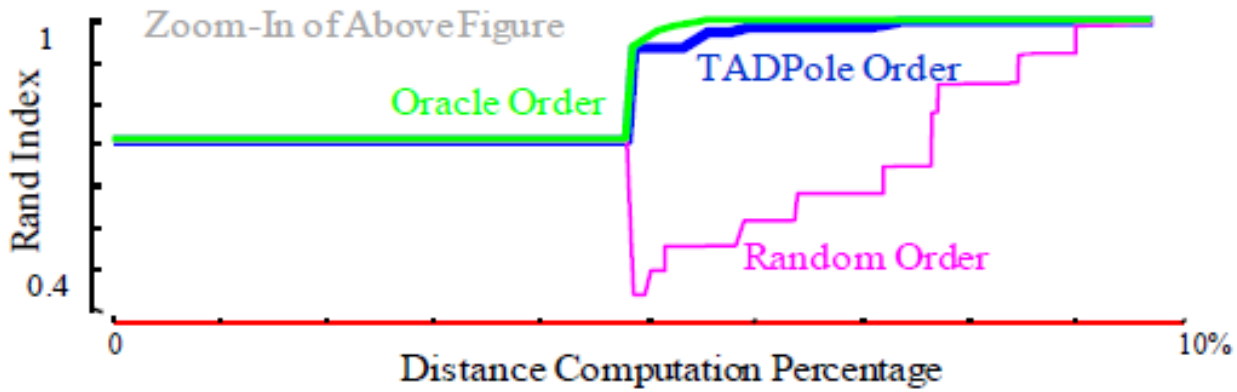




Tadpole clustering

Summary & Results





Tadpole clustering

Summary & Results

- Time series data에 대해 DP를 적용
- Pruning을 적용하여 효율적으로 계산을 수행함
- DTW 사용함에도 불구하고 빠르게 수렴

Table 10: Clustering Quality (in Terms of Rand Index) of TADPole vs. Some State-of-the-Art Clustering Algorithms

Dataset	$TADPole_{DTW}$ ($TADPole_{ED}$)	k-means[10] $DTW_{version}$	Hierarchical $DTW_{version}$	DBSCAN [6] $DTW_{version}$	Spectral [17] $DTW_{version}$
CBF	1 (*0.66*)	0.78	0.73	0.77	0.76
FacesUCR	0.92 (*0.86*)	0.87	0.85	0.77	0.94
MedicalImages	0.66 (*0.67*)	0.67	0.62	0.65	0.69
Symbols	0.98 (*0.81*)	0.93	0.78	0.91	0.95
uWaveGesture_Z	0.86 (*0.84*)	0.85	0.83	0.8	0.86

목차

k-shape clustering

k-Shape: Efficient and Accurate Clustering of Time Series

John Paparrizos
Columbia University
jopa@cs.columbia.edu

Luis Gravano
Columbia University
gravano@cs.columbia.edu

ABSTRACT

The proliferation and ubiquity of temporal data across many disciplines has generated substantial interest in the analysis and mining of time series. Clustering is one of the most popular data mining methods, not only due to its exploratory power, but also as a preprocessing step or subroutine for other techniques. In this paper, we present *k*-Shape, a novel algorithm for time-series clustering. *k*-Shape relies on a scalable iterative refinement procedure, which creates homogeneous and well-separated clusters. As its distance measure, *k*-Shape uses a normalized version of the cross-correlation measure in order to consider the shapes of time series while comparing them. Based on the properties of that distance measure, we develop a method to compute cluster centroids, which are used in every iteration to update the assignment of time series to clusters. To demonstrate the robustness of *k*-Shape, we perform an extensive experimental evaluation of our approach against partitional, hierarchical, and spectral clustering methods, with combinations of the most competitive distance measures. *k*-Shape outperforms all scalable approaches in terms of accuracy. Furthermore, *k*-Shape also

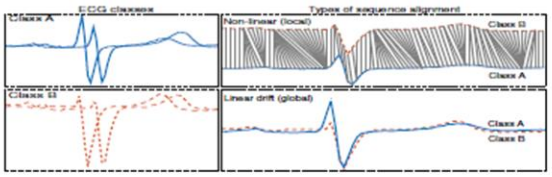


Figure 1: ECG sequence examples and types of alignments for the two classes of the ECGFiveDays dataset [1].

est in querying [2, 19, 45, 46, 48, 62, 74, 79], indexing [9, 13, 41, 42, 44, 77], classification [37, 56, 68, 88], clustering [43, 54, 64, 87, 89], and modeling [4, 38, 86] of such data.

Among all techniques applied to time-series data, clustering is the most widely used as it does not rely on costly human supervision or time-consuming annotation of data. With clustering, we can identify and summarize interesting patterns and correlations in the underlying data [33]. In the last few decades, clustering of time-series sequences has received significant attention [5, 16, 25, 47, 60, 64, 66, 87, 89].

Paparrizos, J., & Gravano, L. (2015, May). k-shape: Efficient and accurate clustering of time series. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data* (pp. 1855-1870). ACM.

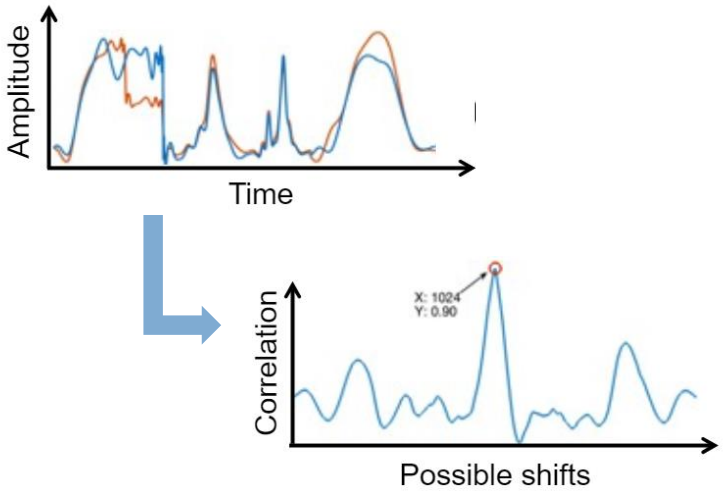
k-shape clustering

summary

- 시계열 데이터가 적합한 K-means를 만들어보자
- Scale-, translate-, and shift-invariant clustering method

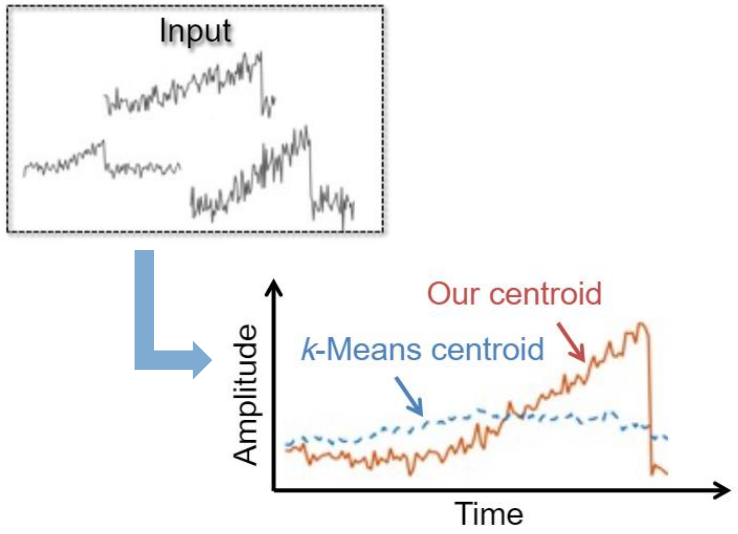
Distance measure

- **A normalized version of cross-correlation measure**



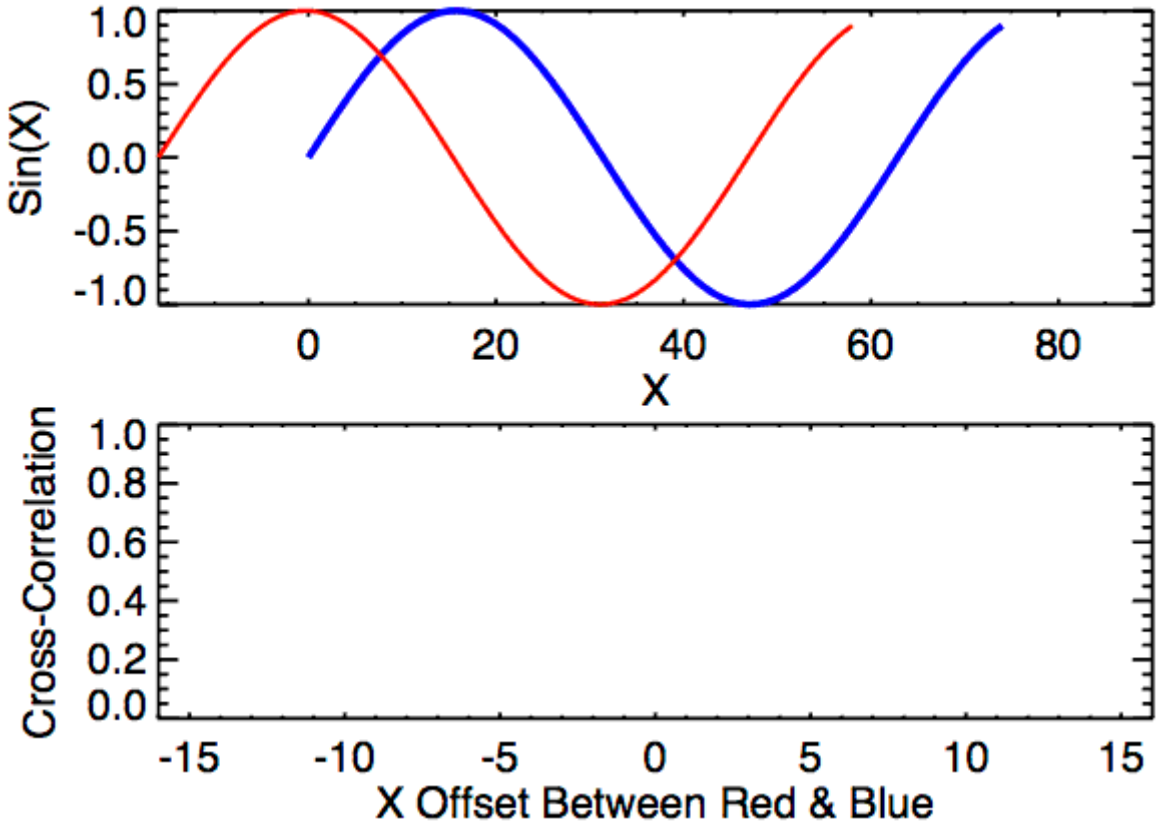
Centroid computations

- **A novel method based on that distance measure**



Distance measure

- Cross-correlation measure
  → **TS1**은 고정을 시켜놓고 **TS2**를 움직이면서 Time lag에 따른 내적값을 산출
  → 두개의 Sequence가 일치하는 점에서 Cross-correlation이 최대값을 보임



Shape-based distance (SBD)

- Shape-based distance

$$SBD(\vec{x},\vec{y}) = 1 - \max_w \left(\frac{CC_w(\vec{x},\vec{y})}{\sqrt{R_0(\vec{x},\vec{x}) \cdot R_0(\vec{y},\vec{y})}} \right)$$

$$-1 \leq \max_w \left(\frac{CC_w(\vec{x},\vec{y})}{\sqrt{R_0(\vec{x},\vec{x}) \cdot R_0(\vec{y},\vec{y})}} \right) \leq 1$$

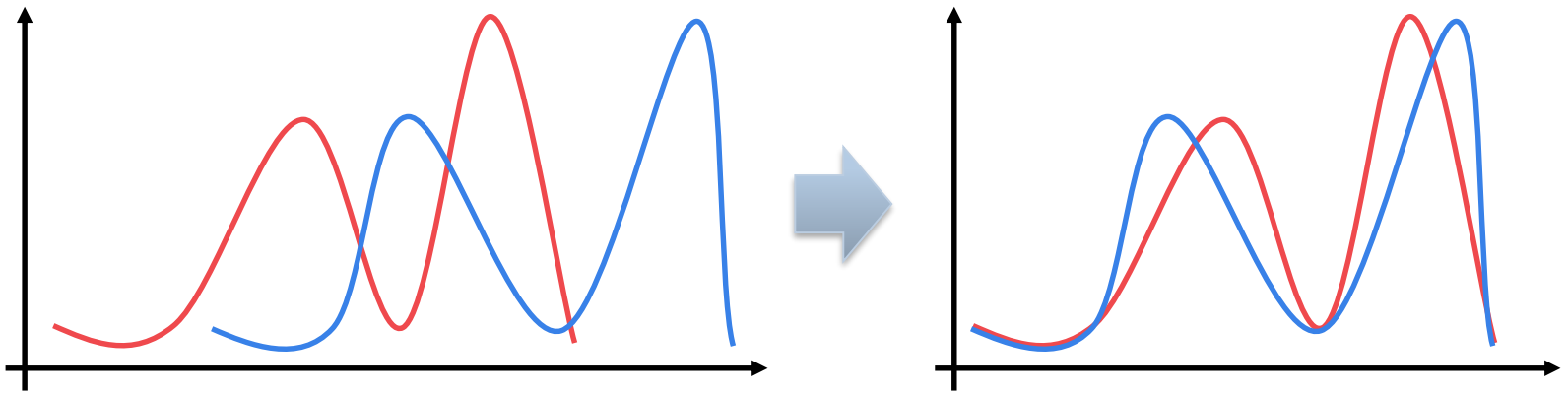
$$0 \leq SBD(\vec{x},\vec{y}) \leq 2$$

- SBD는 0과 2사이의 값을 갖으며, 0에 가까울 수록 두개의 sequence는 유사함

- Fast Fourier Transform(FFT)를 적용함으로써, SBD를 효과적으로 연산할 수 있음

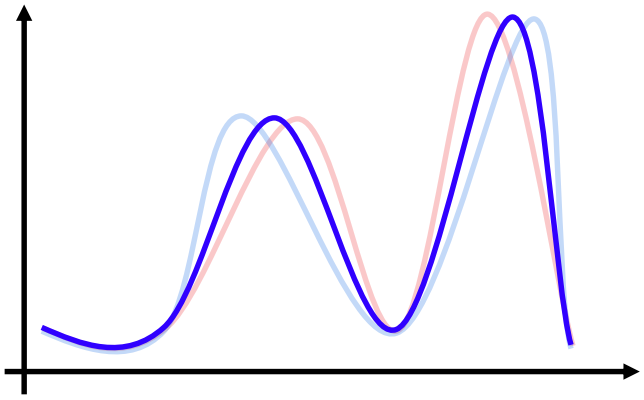
Shape Extraction

- Shape Extraction
  → 1. SBD를 통해 sequence간 시점을 일치 시킴 (= similarity 산출)



Shape Extraction

- Shape Extraction
  → 2. 군집내에서 Sum of squared correlation을 최대화 하도록 군집의 중심을 최적화 식을 통해 산출함



$$\vec{\mu}_k^{\star} = \underset{\vec{\mu}_k}{\operatorname{argmax}} \sum_{\vec{x}_i \in P_k} (\vec{x}_i^T \cdot \vec{\mu}_k)^2$$

$$= \underset{\vec{\mu}_k}{\operatorname{argmax}}\, \vec{\mu_k}^T \cdot \sum_{\vec{x}_i \in P_k} (\vec{x}_i \cdot \vec{x}_i^T) \cdot \vec{\mu_k}$$

$$\vec{\mu_k} = \vec{\mu_k} \cdot Q, \text{ where } Q = I - \frac{1}{m} O$$

$$\vec{\mu_k}^{\star} = \underset{\vec{\mu_k}}{\operatorname{argmax}} \frac{\vec{\mu_k}^T \cdot Q^T \cdot S \cdot Q \cdot \vec{\mu_k}}{\vec{\mu_k}^T \cdot \vec{\mu_k}}$$

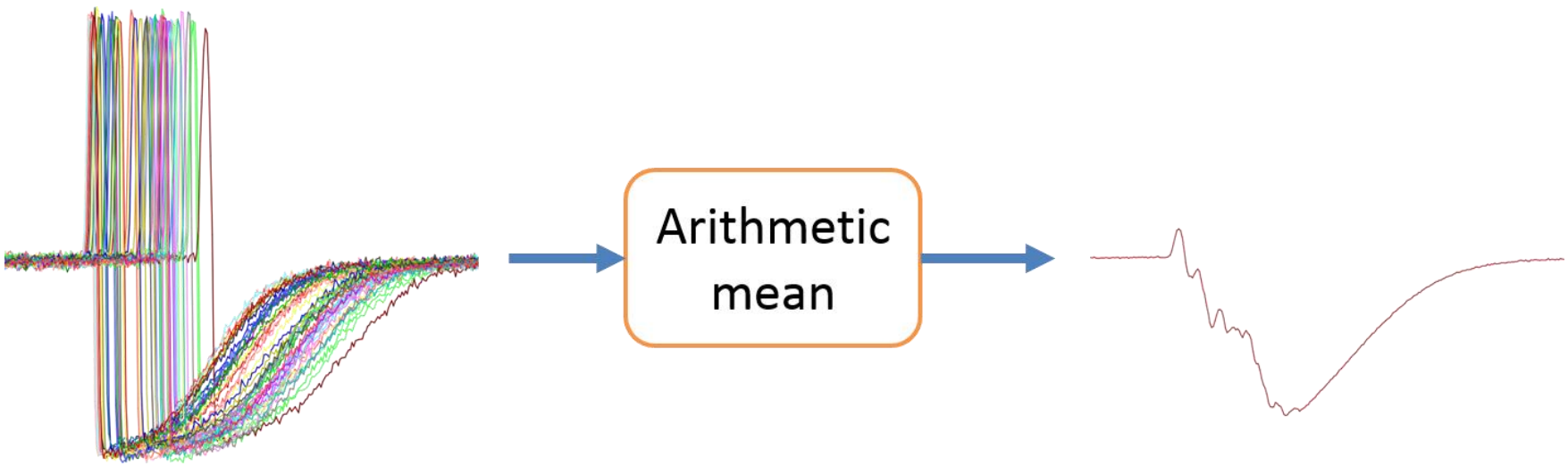
$$= \underset{\vec{\mu_k}}{\operatorname{argmax}} \frac{\vec{\mu_k}^T \cdot M \cdot \vec{\mu_k}}{\vec{\mu_k}^T \cdot \vec{\mu_k}}$$

$\vec{\mu_k}^{\star}$ eigenvector that corresponds to the larest eigenvalue

Rayleigh Quotient

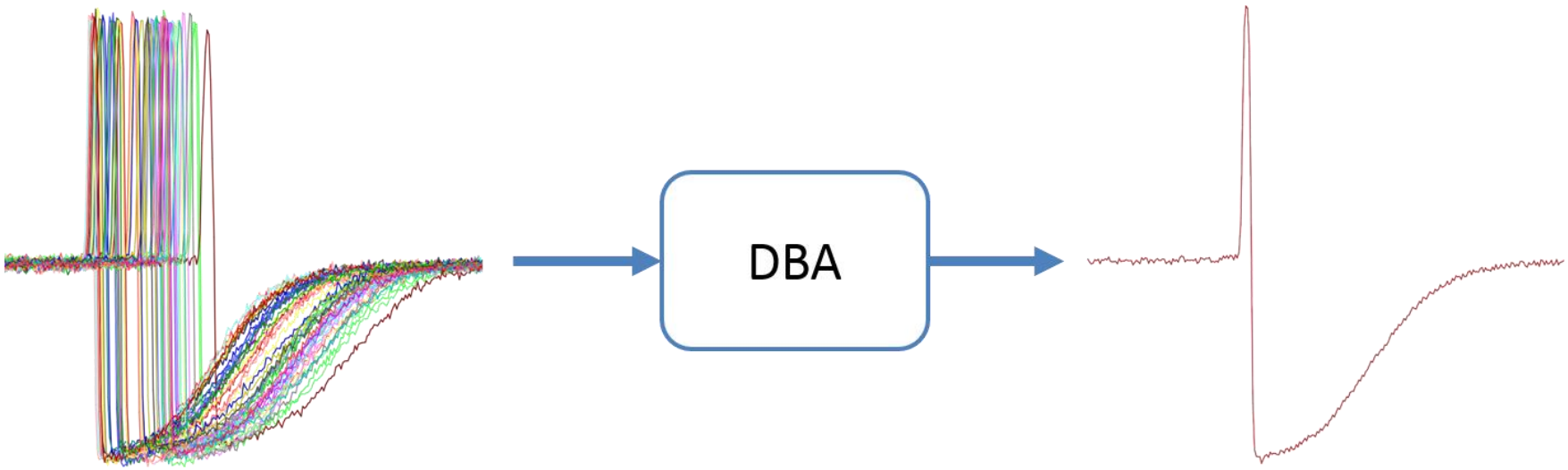
(참고)Extract centroid #1

- Arithmetic mean



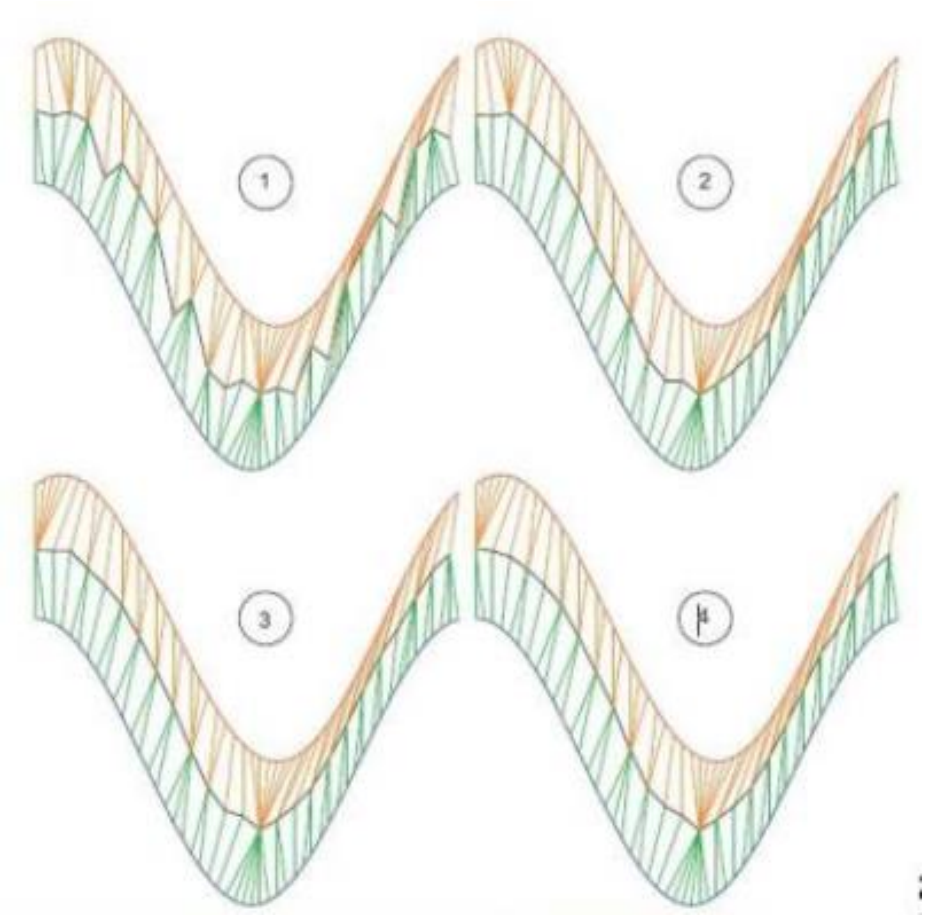
(참고)Extract centroid #2

- DTW barycenter averaging (DBA)



(참고) Extract centroid #2

- DTW barycenter averaging (DBA)



K-shape clustering

- K-means와 동일한 프로세스로 작동

Algorithm 3: $[IDX, C] = k\text{-}Shape(X, k)$

Input: X is an n-by-m matrix containing n time series of length m that are initially z-normalized.
k is the number of clusters to produce.

Output: IDX is an n-by-1 vector containing the assignment of n time series to k clusters (initialized randomly).
C is a k-by-m matrix containing k centroids of length m (initialized as vectors with all zeros).

```
1  iter ← 0
2  IDX' ← [ ]
3  while IDX != IDX' and iter < 100 do
4      IDX' ← IDX
       // Refinement step
5      for j ← 1 to k do
6          X' ← [ ]
7          for i ← 1 to n do
8              if IDX(i) = j then
9                  X' ← [X'; X(i)]
10         C(j) ← ShapeExtraction(X', C(j))        // Algorithm 2
       // Assignment step
11     for i ← 1 to n do
12         mindist ← ∞
13         for j ← 1 to k do
14             [dist, x'] ← SBD(C(j), X(i))        // Algorithm 1
15             if dist < mindist then
16                 mindist ← dist
17                 IDX(i) ← j
18     iter ← iter + 1
```

Results

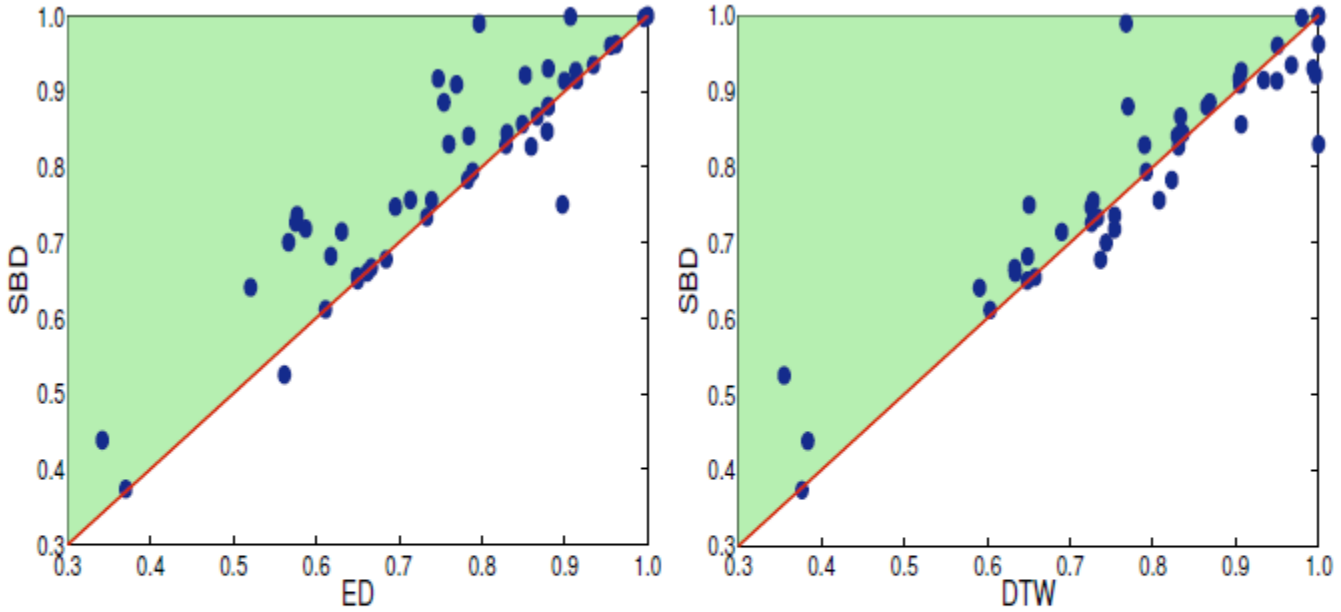
Evaluation of SBD

Distance Measure	>	=	<	Better	Average Accuracy	Runtime
DTW DTW_{LB}	29	2	17	✓	0.788	15573x 6040x
$cDTW^{opt}$ $cDTW^{opt}_{LB}$	31	15	2	✓	0.814	2873x 322x
$cDTW^{5}$ $cDTW^{5}_{LB}$	34	3	11	✓	0.809	1558x 122x
$cDTW^{10}$ $cDTW^{10}_{LB}$	33	1	14	✓	0.804	2940x 364x
SBD_{NoFFT} SBD_{NoPow2} SBD	30	12	6	✓	0.795	224x 8.7x 4.4x

Table 2: Comparison of distance measures. Columns ">", "=", and "<" denote the number of datasets over which a distance measure is better, equal, or worse, respectively, in comparison to ED. "Better" indicates that a distance measure outperforms ED with statistical significance. "Average accuracy" denotes the accuracy achieved in the 48 datasets whereas "Runtime" indicates the factor by which a distance measure is slower than ED.

Results

Evaluation of SBD



(a) SBD vs. ED (b) SBD vs. DTW

Figure 5: Comparison of SBD, ED, and DTW over 48 datasets. Circles above the diagonal indicate datasets over which SBD has better accuracy than the compared method.

Results

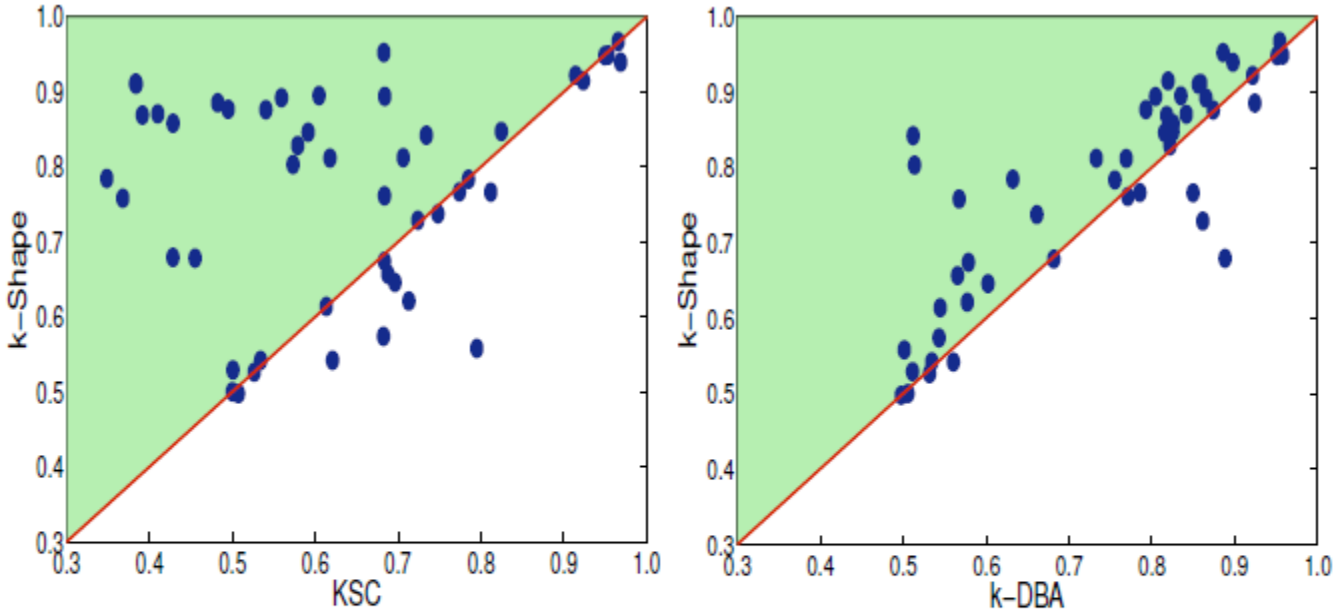
Evaluation of k-shape

Algorithm	>	=	<	Better	Worse	Rand Index	Runtime
k-AVG+SBD	32	1	15	✗	✗	0.745	3.6x
k-AVG+DTW	10	0	38	✗	✓	0.584	3444x
KSC	22	0	26	✗	✗	0.636	448x
k-DBA	18	0	30	✗	✗	0.733	3892x
k-Shape+DTW	19	1	28	✗	✗	0.698	4175x
k-Shape	36	1	11	✓	✗	0.772	12.4x

Table 3: Comparison of k-means variants against k-AVG+ED. Columns ">", "=", and "<" denote the number of datasets over which an algorithm is better, equal, or worse, respectively, in comparison to k-AVG+ED. "Better" indicates that an algorithm outperforms k-AVG+ED with statistical significance whereas "Worse" indicates that k-AVG+ED outperforms an algorithm with statistical significance. "Rand Index" denotes the accuracy achieved in the 48 datasets whereas "Runtime" indicates the factor by which an algorithm is slower than k-AVG+ED.

Results

Evaluation of k-shape



(a) k-Shape vs. KSC (b) k-Shape vs. k-DBA

Figure 7: Comparison of k-Shape, KSC, and k-DBA over 48 datasets. Circles above the diagonal indicate datasets over which k-Shape has better Rand Index.

KSC: K-Spectral Centroid

Yang, J., & Leskovec, J. (2011, February). Patterns of temporal variation in online media. In *Proceedings of the fourth ACM international conference on Web search and data mining* (pp. 177-186). ACM.

감사합니다