Dynamic Time Warping

Farhan Tahmidul Karim[1] Md Ishrak Ahsan[2]
Riad Ahmed Anonto[3]

[1]1905035

[2]1905045

[3]1905050

[1,2,3]Department of Computer Science and Technology,BUET

What is Dynamic Time Warping?

Dynamic Time Warping is

- An algorithm used for measuring the similarity between two temporal time series sequence
- Computes the distance from the matching similar elements between two series
- Used in dynamic programming to find the optimal path

What is Dynamic Time Warping?

***Dynamic Time Warping* is**

- An algorithm used for measuring the similarity between two temporal time series sequence
- Computes the distance from the matching similar elements between two series
- Used in dynamic programming to find the optimal path

What is Dynamic Time Warping?

***Dynamic Time Warping* is**

- An algorithm used for measuring the similarity between two temporal time series sequence
- Computes the distance from the matching similar elements between two series
- Used in dynamic programming to find the optimal path

Motivation for DTW

Tells us basically two things :

- How similar are two signals

Motivation for DTW

Tells us basically two things :

- How similar are two signals
- Which points correspond to one another

So How do you figure out how similar the Signals are?

First Approach

Euclidean Matching :

Compare the Signals point by point

In fact, this approach is called the "Naive Approach"

So How do you figure out how similar the Signals are?

First Approach

Euclidean Matching :

Compare the Signals point by point

In fact, this approach is called the "Naive Approach"

So How do you figure out how similar the Signals are?

Second Approach

Dynamic Time Warping :

Used for two variable-length arrays or time sequences to create the best possible alignment.

Exploiting the temporal distortions between them.

So How do you figure out how similar the Signals are?

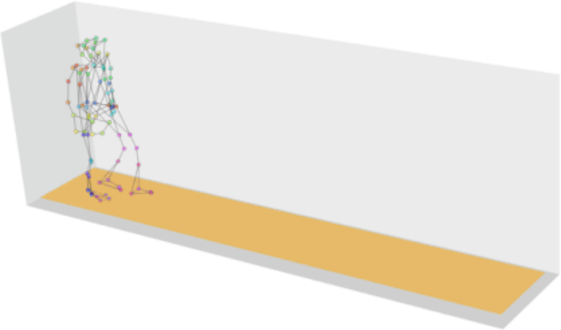
Second Approach

Dynamic Time Warping :

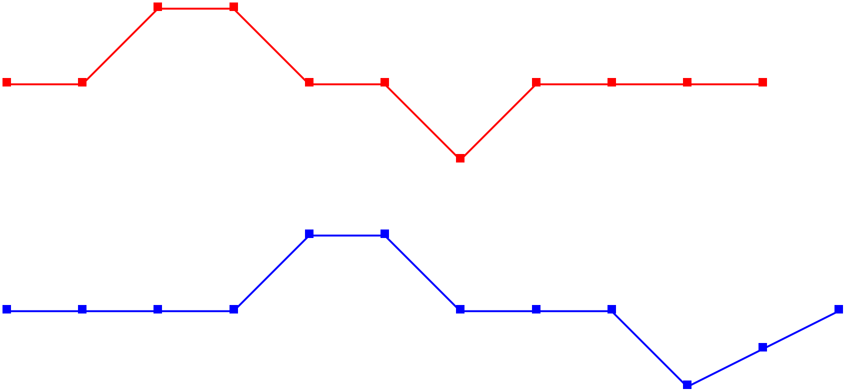
Used for two variable-length arrays or time sequences to create the best possible alignment.

Exploiting the temporal distortions between them.

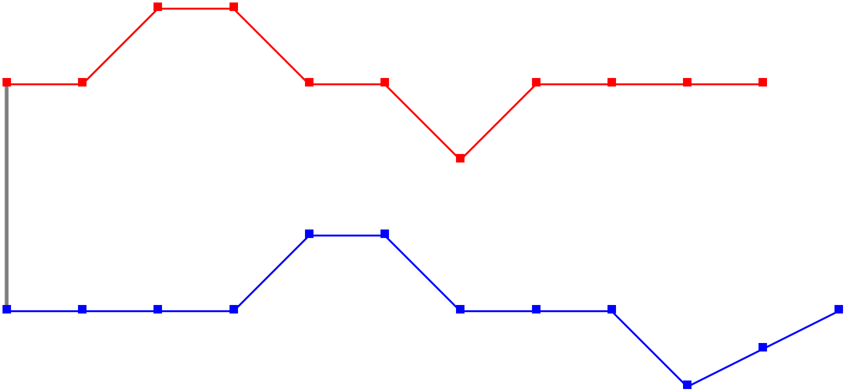
Similarity Detection



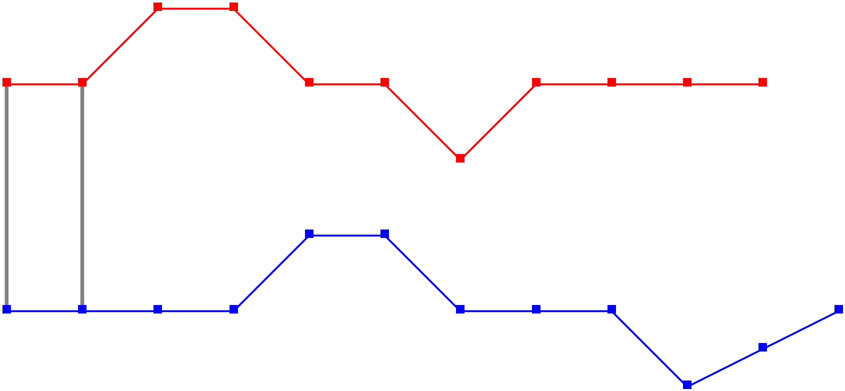
Euclidean Matching Vs Dynamic Time Warping



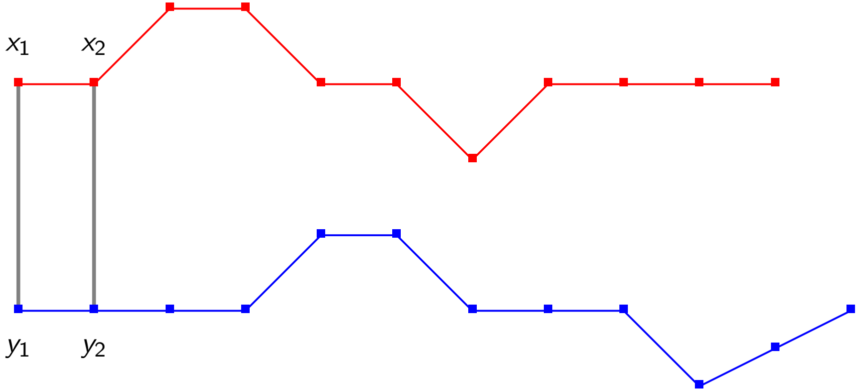
Euclidean Matching Vs Dynamic Time Warping



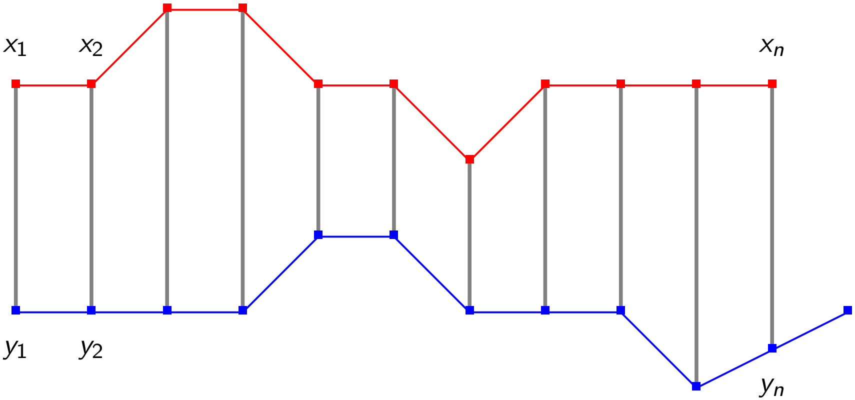
Euclidean Matching Vs Dynamic Time Warping



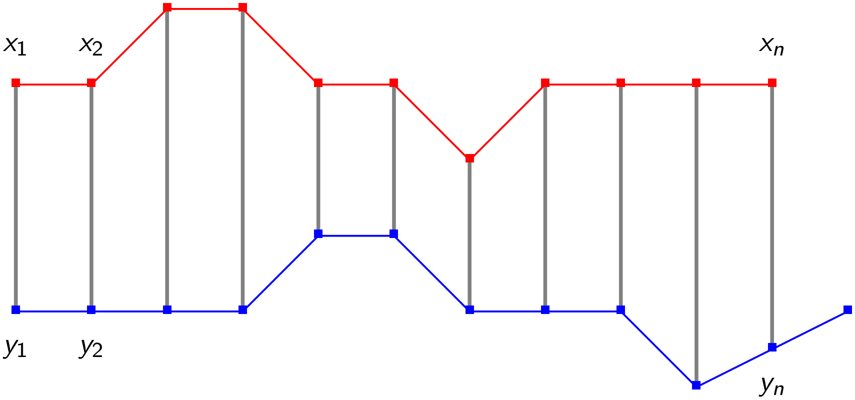
Euclidean Matching Vs Dynamic Time Warping



Euclidean Matching Vs Dynamic Time Warping



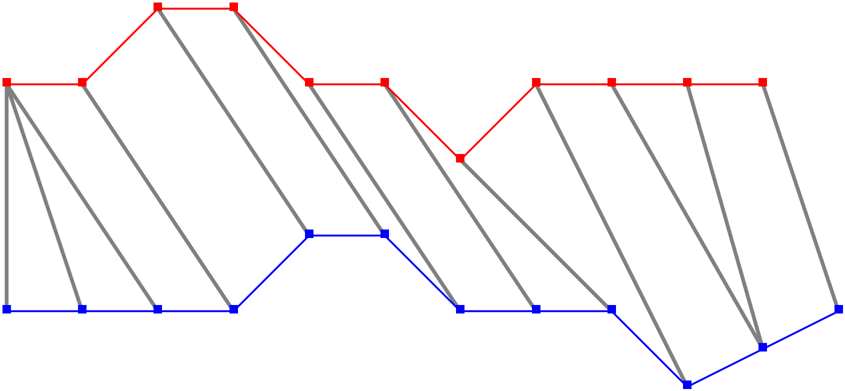
Euclidean Matching Vs Dynamic Time Warping



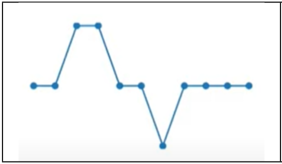
Formula can be written as :

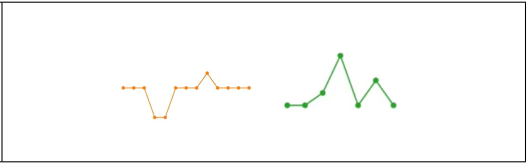
$$d(x_{1:N}, y_{1:M}) = \sum^{n} |x_i - y_i| \quad (1)$$

Euclidean Matching Vs Dynamic Time Warping



Dynamic Time Warping





How DTW is Different?

Euclidean Matching

- ✓ One to one point Comparison
- ✓ That's why compares only time series of same length

Dynamic Time Warping

- ✓ Allows many-to-one comparisons
- ✓ Time series of different length can be compared

Algorithm

```
int DTWDistance(s: array [1..n], t: array [1..m]) {
   DTW := array [0..n, 0..m]
   for i := 0 to n
      for j := 0 to m
         DTW[i, j] := ∞
   DTW[0, 0] := 0
   for i := 1 to n
      for j := 1 to m
         cost := d(s[i], t[j])
         DTW[i, j] := cost + min(DTW[i-1, j], // insertion
                                 DTW[i, j-1], // deletion
                                 DTW[i-1, j-1]) // match
   return DTW[n, m]
}
```

Distance Matrix

Let us consider two time series

$$TS_A = [1, 3, 4, 9, 8, 2]$$

$$TS_B = [1, 6, 2, 3, 0, 9, 4]$$

Distance Matrix

Let us consider two time series

$$TS_A = [1, 3, 4, 9, 8, 2]$$

$$TS_B = [1, 6, 2, 3, 0, 9, 4]$$

Distance Matrix

2	21	13	9	10	12	16	11
8	20	9	13	16	19	9	12
9	13	7	11	11	14	8	13
4	5	4	5	5	8	12	12
3	2	3	4	4	7	12	14
1	0	5	6	8	9	17	20
	1	6	2	3	0	9	4

$$D_{i,j} = |A_i - B_j| + min(D_{i-1,j-1}, D_{i-1,j}, D_{i,j-1})$$

Distance Matrix

2	21	13	9	10	12	16	11
8	20	9	13	16	19	9	12
9	13	7	11	11	14	8	13
4	5	4	5	5	8	12	12
3	2	3	4	4	7	12	14
1	0	5	6	8	9	17	20
	1	6	2	3	0	9	4

$$D_{i,j} = |A_i - B_j| + min(D_{i-1,j-1}, D_{i-1,j}, D_{i,j-1})$$

Distance Matrix

2	21	13	9	10	12	16	11
8	20	9	13	16	19	9	12
9	13	7	11	11	14	8	13
4	5	4	5	5	8	12	12
3	2	3	4	4	7	12	14
1	0	5	6	8	9	17	20
	1	6	2	3	0	9	4

$$D_{i,j} = |A_i - B_j| + min(D_{i-1,j-1}, D_{i-1,j}, D_{i,j-1})$$

Distance Matrix

2	21	13	9	10	12	16	11
8	20	9	13	16	19	9	12
9	13	7	11	11	14	8	13
4	5	4	5	5	8	12	12
3	2	3	4	4	7	12	14
1	0	5	6	8	9	17	20
	1	6	2	3	0	9	4

$$D_{i,j} = |A_i - B_j| + min(D_{i-1,j-1}, D_{i-1,j}, D_{i,j-1})$$

Distance Matrix

2	21	13	9	10	12	16	11
8	20	9	13	16	19	9	12
9	13	7	11	11	14	8	13
4	5	4	5	5	8	12	12
3	2	3	4	4	7	12	14
1	0	5	6	8	9	17	20
	1	6	2	3	0	9	4

$$D_{i,j} = |A_i - B_j| + min(D_{i-1,j-1}, D_{i-1,j}, D_{i,j-1})$$

Distance Matrix

2	21	13	9	10	12	16	11
8	20	9	13	16	19	9	12
9	13	7	11	11	14	8	13
4	5	4	5	5	8	12	12
3	2	3	4	4	7	12	14
1	0	5	6	8	9	17	20
	1	6	2	3	0	9	4

$$D_{i,j} = |A_i - B_j| + min(D_{i-1,j-1}, D_{i-1,j}, D_{i,j-1})$$

Distance Matrix

2	21	13	9	10	12	16	11
8	20	9	13	16	19	9	12
9	13	7	11	11	14	8	13
4	5	4	5	5	8	12	12
3	2	3	4	4	7	12	14
1	0	5	6	8	9	17	20
	1	6	2	3	0	9	4

$$D_{i,j} = |A_i - B_j| + min(D_{i-1,j-1}, D_{i-1,j}, D_{i,j-1})$$

Distance Matrix

2	21	13	9	10	12	16	11
8	20	9	13	16	19	9	12
9	13	7	11	11	14	8	13
4	5	4	5	5	8	12	12
3	2	3	4	4	7	12	14
1	0	5	6	8	9	17	20
	1	6	2	3	0	9	4

$$D_{i,j} = |A_i - B_j| + min(D_{i-1,j-1}, D_{i-1,j}, D_{i,j-1})$$

Distance Matrix

2	21	13	9	10	12	16	11
8	20	9	13	16	19	9	12
9	13	7	11	11	14	8	13
4	5	4	5	5	8	12	12
3	2	3	4	4	7	12	14
1	0	5	6	8	9	17	20
	1	6	2	3	0	9	4

$$D_{i,j} = |A_i - B_j| + min(D_{i-1,j-1}, D_{i-1,j}, D_{i,j-1})$$

Distance Matrix

2	21	13	9	10	12	16	11
8	20	9	13	16	19	9	12
9	13	7	11	11	14	8	13
4	5	4	5	5	8	12	12
3	2	3	4	4	7	12	14
1	0	5	6	8	9	17	20
	1	6	2	3	0	9	4

$$D_{i,j} = |A_i - B_j| + min(D_{i-1,j-1}, D_{i-1,j}, D_{i,j-1})$$

Distance Matrix

2	21	13	9	10	12	16	11
8	20	9	13	16	19	9	12
9	13	7	11	11	14	8	13
4	5	4	5	5	8	12	12
3	2	3	4	4	7	12	14
1	0	5	6	8	9	17	20
	1	6	2	3	0	9	4

$$D_{i,j} = |A_i - B_j| + min(D_{i-1,j-1}, D_{i-1,j}, D_{i,j-1})$$

Distance Matrix

2	21	13	9	10	12	16	11
8	20	9	13	16	19	9	12
9	13	7	11	11	14	8	13
4	5	4	5	5	8	12	12
3	2	3	4	4	7	12	14
1	0	5	6	8	9	17	20
	1	6	2	3	0	9	4

$$D_{i,j} = |A_i - B_j| + min(D_{i-1,j-1}, D_{i-1,j}, D_{i,j-1})$$

Distance Matrix

2	21	13	9	10	12	16	11
8	20	9	13	16	19	9	12
9	13	7	11	11	14	8	13
4	5	4	5	5	8	12	12
3	2	3	4	4	7	12	14
1	0	5	6	8	9	17	20
	1	6	2	3	0	9	4

$$D_{i,j} = |A_i - B_j| + min(D_{i-1,j-1}, D_{i-1,j}, D_{i,j-1})$$

Optimal Path

2	21	13	9	10	12	16	11
8	20	9	13	16	19	9	12
9	13	7	11	11	14	8	13
4	5	4	5	5	8	12	12
3	2	3	4	4	7	13	14
1	0	5	6	8	9	17	20
	1	6	2	3	0	9	4

1. Start at $D_{n,m}$
2. Find $min(D_{i-1,j-1}, D_{i-1,j}, D_{i,j-1})$, add to current path and jump to that cell
3. Proceed until we reach $D_{1,1}$

Optimal Path

2	21	13	9	10	12	16	11
8	20	9	13	16	19	9	12
9	13	7	11	11	14	8	13
4	5	4	5	5	8	12	12
3	2	3	4	4	7	13	14
1	0	5	6	8	9	17	20
	1	6	2	3	0	9	4

1. Start at $D_{n,m}$
2. Find $min(D_{i-1,j-1}, D_{i-1,j}, D_{i,j-1})$, add to current path and jump to that cell
3. Proceed until we reach $D_{1,1}$

Optimal Path

2	21	13	9	10	12	16	11
8	20	9	13	16	19	9	12
9	13	7	11	11	14	8	13
4	5	4	5	5	8	12	12
3	2	3	4	4	7	13	14
1	0	5	6	8	9	17	20
	1	6	2	3	0	9	4

1. Start at $D_{n,m}$
2. Find $min(D_{i-1,j-1}, D_{i-1,j}, D_{i,j-1})$, add to current path and jump to that cell
3. Proceed until we reach $D_{1,1}$

Optimal Path

2	21	13	9	10	12	16	11
8	20	9	13	16	19	9	12
9	13	7	11	11	14	8	13
4	5	4	5	5	8	12	12
3	2	3	4	4	7	13	14
1	0	5	6	8	9	17	20
	1	6	2	3	0	9	4

1. Start at $D_{n,m}$
2. Find $min(D_{i-1,j-1}, D_{i-1,j}, D_{i,j-1})$, add to current path and jump to that cell
3. Proceed until we reach $D_{1,1}$

Optimal Path

2	21	13	9	10	12	16	11
8	20	9	13	16	19	9	12
9	13	7	11	11	14	8	13
4	5	4	5	5	8	12	12
3	2	3	4	4	7	13	14
1	0	5	6	8	9	17	20
	1	6	2	3	0	9	4

1. Start at $D_{n,m}$
2. Find $min(D_{i-1,j-1}, D_{i-1,j}, D_{i,j-1})$, add to current path and jump to that cell
3. Proceed until we reach $D_{1,1}$

Optimal Path

2	21	13	9	10	12	16	11
8	20	9	13	16	19	9	12
9	13	7	11	11	14	8	13
4	5	4	5	5	8	12	12
3	2	3	4	4	7	13	14
1	0	5	6	8	9	17	20
	1	6	2	3	0	9	4

1. Start at $D_{n,m}$
2. Find $min(D_{i-1,j-1}, D_{i-1,j}, D_{i,j-1})$, add to current path and jump to that cell
3. Proceed until we reach $D_{1,1}$

Optimal Path

2	21	13	9	10	12	16	11
8	20	9	13	16	19	9	12
9	13	7	11	11	14	8	13
4	5	4	5	5	8	12	12
3	2	3	4	4	7	13	14
1	0	5	6	8	9	17	20
	1	6	2	3	0	9	4

1. Start at $D_{n,m}$
2. Find $min(D_{i-1,j-1}, D_{i-1,j}, D_{i,j-1})$, add to current path and jump to that cell
3. Proceed until we reach $D_{1,1}$

Optimal Path

2	21	13	9	10	12	16	11
8	20	9	13	16	19	9	12
9	13	7	11	11	14	8	13
4	5	4	5	5	8	12	12
3	2	3	4	4	7	13	14
1	0	5	6	8	9	17	20
	1	6	2	3	0	9	4

1. Start at $D_{n,m}$
2. Find $min(D_{i-1,j-1}, D_{i-1,j}, D_{i,j-1})$, add to current path and jump to that cell
3. Proceed until we reach $D_{1,1}$

Optimal Path

2	21	13	9	10	12	16	11
8	20	9	13	16	19	9	12
9	13	7	11	11	14	8	13
4	5	4	5	5	8	12	12
3	2	3	4	4	7	13	14
1	0	5	6	8	9	17	20
	1	6	2	3	0	9	4

1. Start at $D_{n,m}$
2. Find $min(D_{i-1,j-1}, D_{i-1,j}, D_{i,j-1})$, add to current path and jump to that cell
3. Proceed until we reach $D_{1,1}$

Applications of DTW

- Spoken Word Recognition
- Detect Sales & Trend
- Wearable Fitness Trackers
- Route and ETA Calculation

Applications of DTW

- Spoken Word Recognition
- Detect Sales & Trend
- Wearable Fitness Trackers
- Route and ETA Calculation

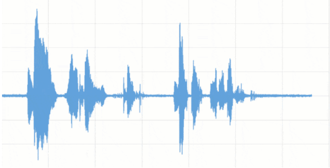
Applications of DTW

- Spoken Word Recognition
- Detect Sales & Trend
- Wearable Fitness Trackers
- Route and ETA Calculation

Applications of DTW

- Spoken Word Recognition
- Detect Sales & Trend
- Wearable Fitness Trackers
- Route and ETA Calculation

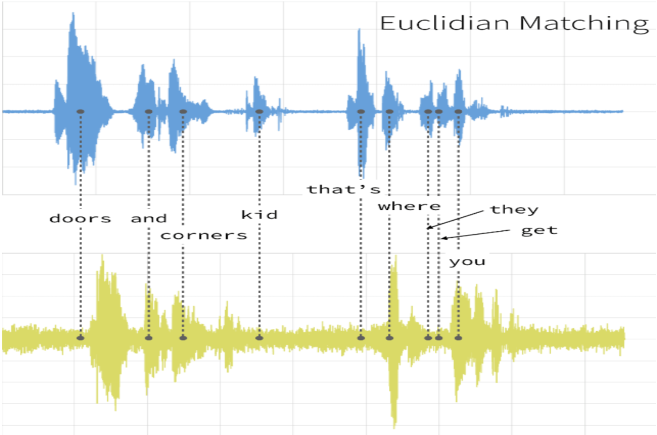
Spoken Word Recognition by Matching Sound Pattern



"Doors and corners, kid. That's where they get you v1

"Doors and corners, kid. That's where they get you v2

Spoken Word Recognition by Sound Matching Pattern: Euclidian Approach



Failure of Euclidian matching in identifying speech delays/pauses

Spoken Word Recognition by Sound Matching Pattern: Using DTW

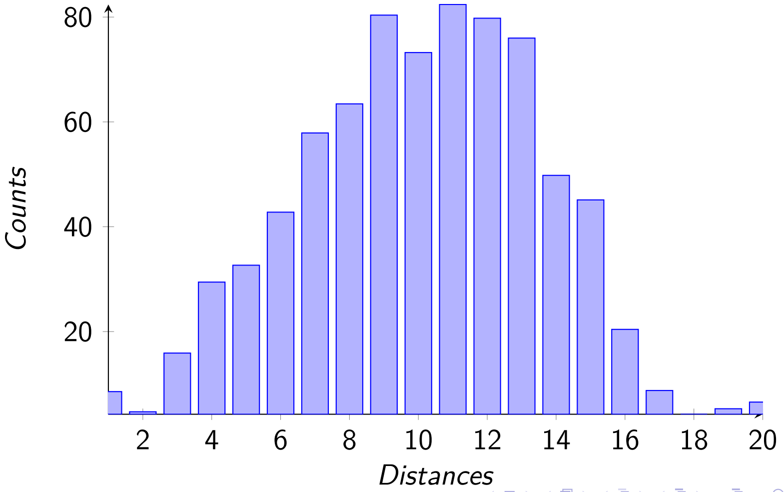
Detecting Sales Trends

Units of each product sold per week				
Product Code	**W1**	**W2**	**...**	**W52**
P1	11	12	...	7
P2	7	6	...	4
P3	7	11	...	3
P4	12	8	...	2
P5	8	5	...	0
P6	4	3	...	10
P7	5	6	...	5
...	...	...	...	...
P20	18	6	...	7

Weekly sales transaction data set of a company throughout last year

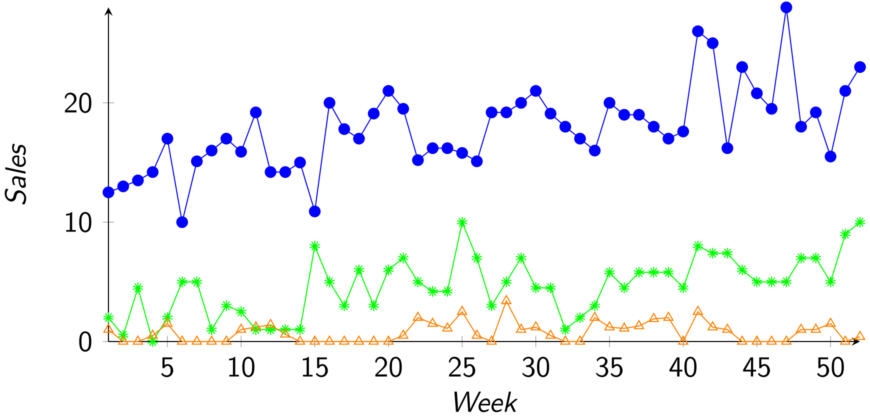
Detecting Sales Trends

DTW distances for each pairwise product sales comparison



Detecting Sales Trends

Comparing Optimal Sales Trend with the Furthest and Closest Produ




Any
Question