

Time Series Searching, Forecasting, and Classification with Applications in Bioinformatics

時系列の探索・予測・分類とその生命情報学への応用

Coleman Yu

余 浩旻

Graduated from HKUST.

I am from Hong Kong.

cyu@kuicr.kyoto-u.ac.jp

Pre-defense (20-01-2026)



Outline

- Introduction
- 1. MTSCCleav: a Multivariate **Time Series Classification** (MTSC)-based Method for Predicting Human Dicer Cleavage
- 2. Scaling with Multiple Scaling Factors in **Time Series Searching**
 - Distance Measure
- 3. Leveraging Nearest Neighbors for **Time Series Forecasting** with Matrix Profile
- Conclusion
 - A data mining primitive

Bioinformatics

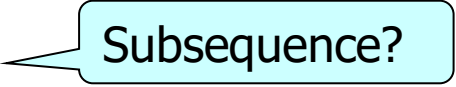




Outline

- Introduction
- 1. MTSCCleav: a Multivariate Time Series Classification (MTSC)-based Method for Predicting Human Dicer Cleavage
- 2. Scaling with Multiple Scaling Factors in Time Series Searching
- 3. Leveraging Nearest Neighbors for Time Series Forecasting with Matrix Profile
- Conclusion



Introduction

- What is Time Series? 
- Time Series is everywhere! 
- Tasks on Time Series 

What is Time Series?

A measurement

Behavioral attribute

A Real Number

A Time Series $T = t_1, t_2, \dots, t_n$
with length = n

Uniform sampling rate

A timestamp

Contextual attribute

A discrete index

■ $T = 1, 0, -1, 1.1$ ✓

■ $T = A, G, C, T$ ✗

String

■ $T = 1, 2, 3, 4$ ✓

Alphabet

Everywhere!

- 1. Stock price



- 2. Motion data



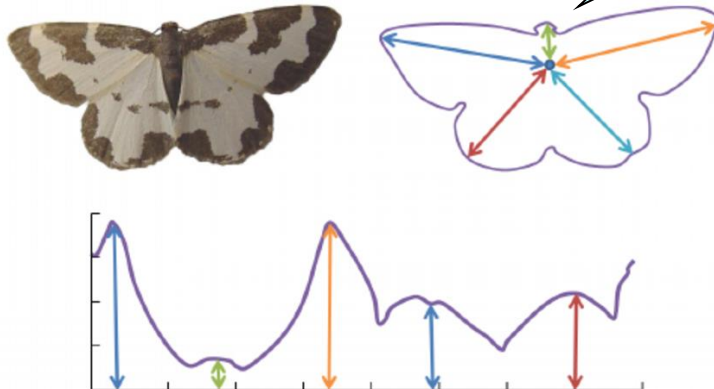
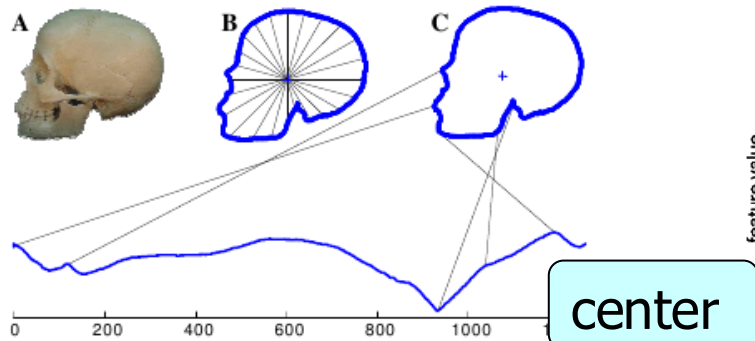
- 3. ECG, EEG



And More! Transformation!

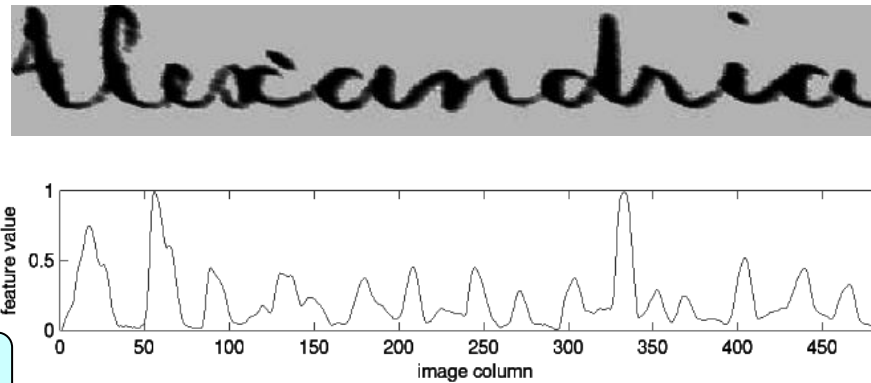
4. Satellite Image Time Series (SITS)

1. Angle-based method



contour

2. Linear scanning



3. Image to Time series

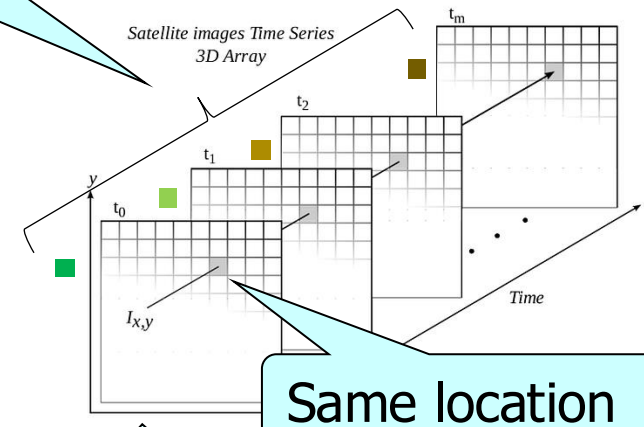


1D array of pixels

1D array of pixels

⋮

Deforestation

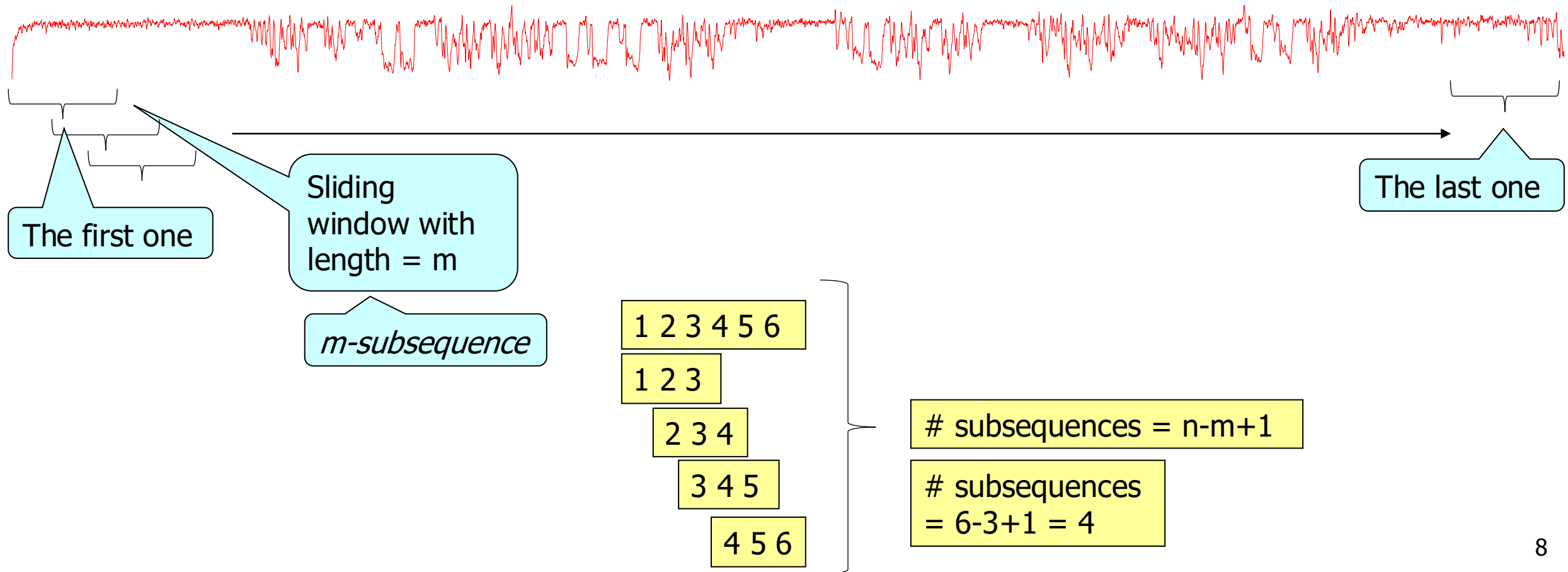


An image

Flattening

Subsequences

Definition (Subsequence). A subsequence $T(i : j)$ of a time series T is a shorter time series that starts from position i and ends at position j with length $= j - i + 1$. Both ends are inclusive. Formally, $T(i : j) = t_i, t_{i+1}, \dots, t_j$, $1 \leq i \leq j \leq n$.

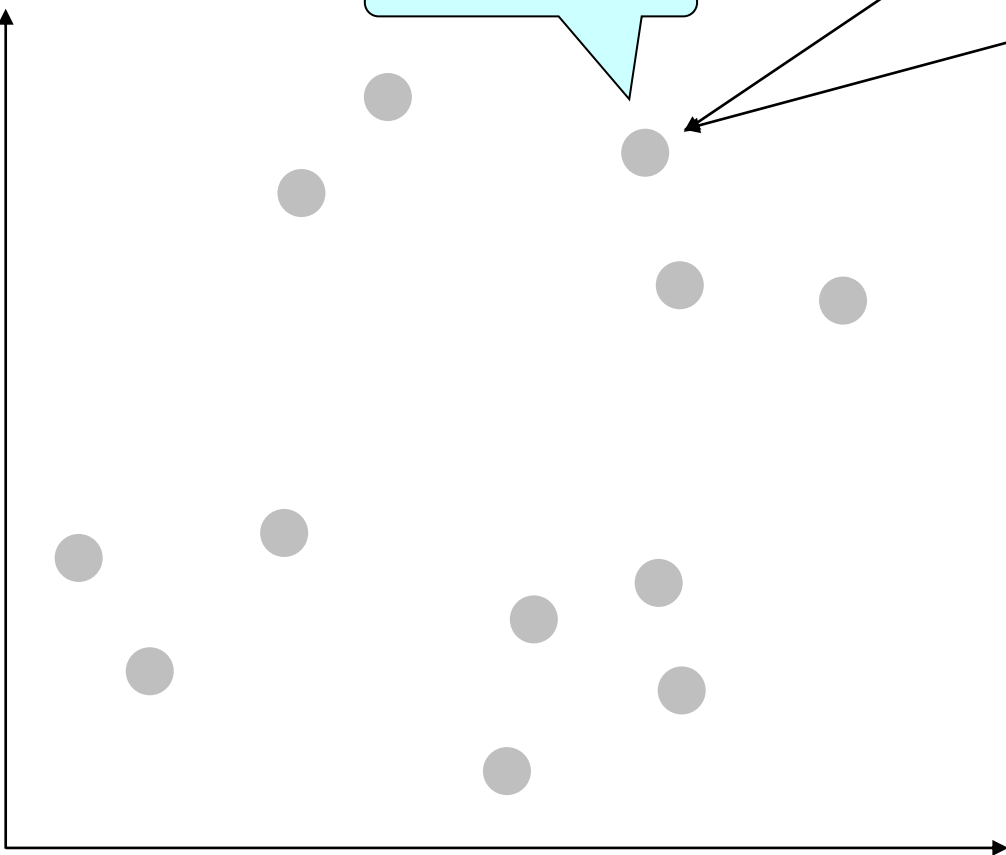


Tasks

A Time series T with length n

A Time series

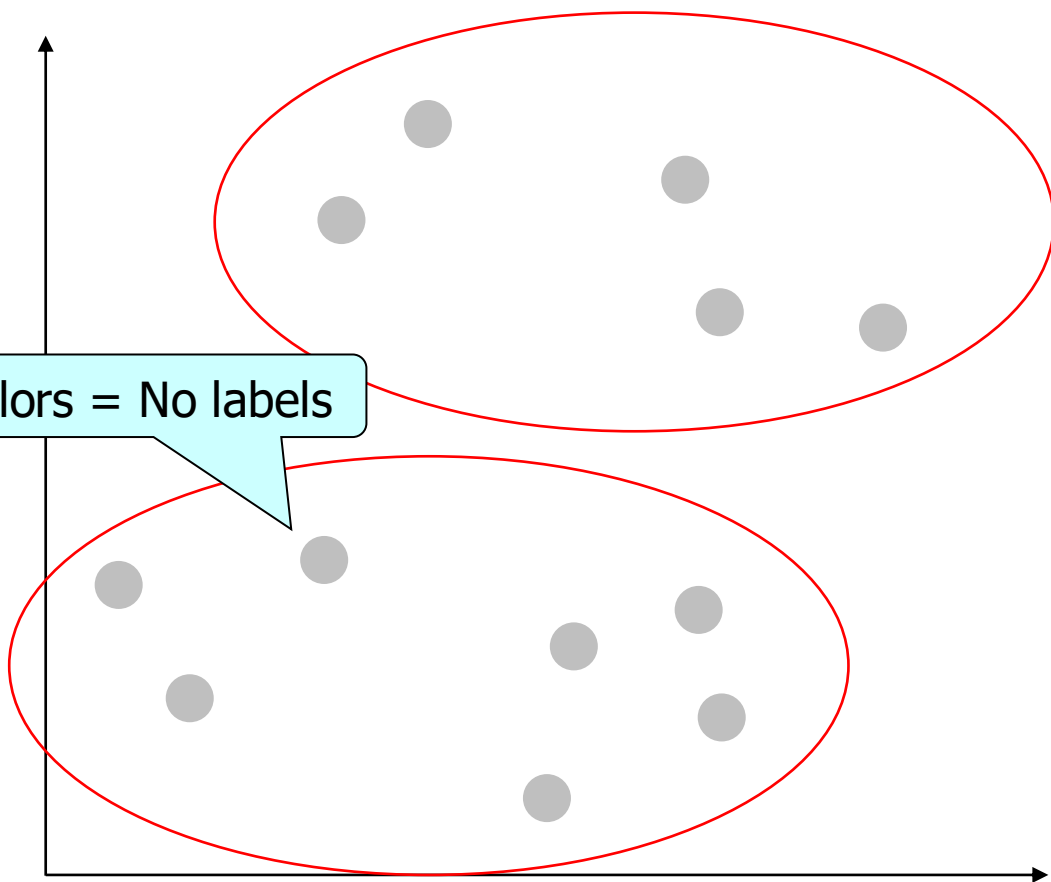
View it as a point in n -dimensional space



Task

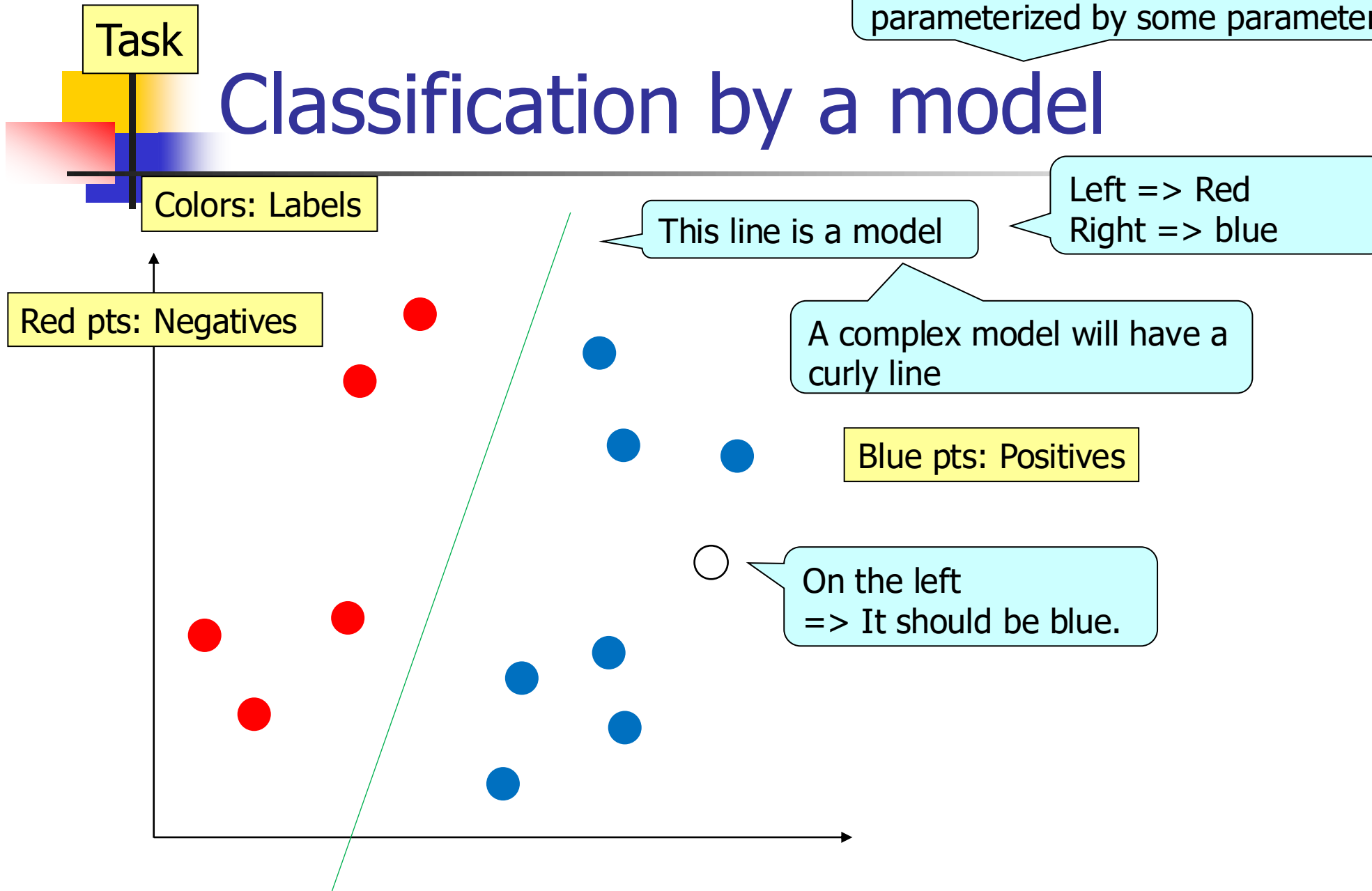
Clustering

No colors = No labels



Classification by a model

We train on the data to obtain a model parameterized by some parameters.



No higher level has been learnt.

This method is "lazy".

A famous classifier in the field of time series

Task

Classification by Nearest Neighbor (NN)

Colors: Labels

If k -NN, majority vote.

It should be blue.

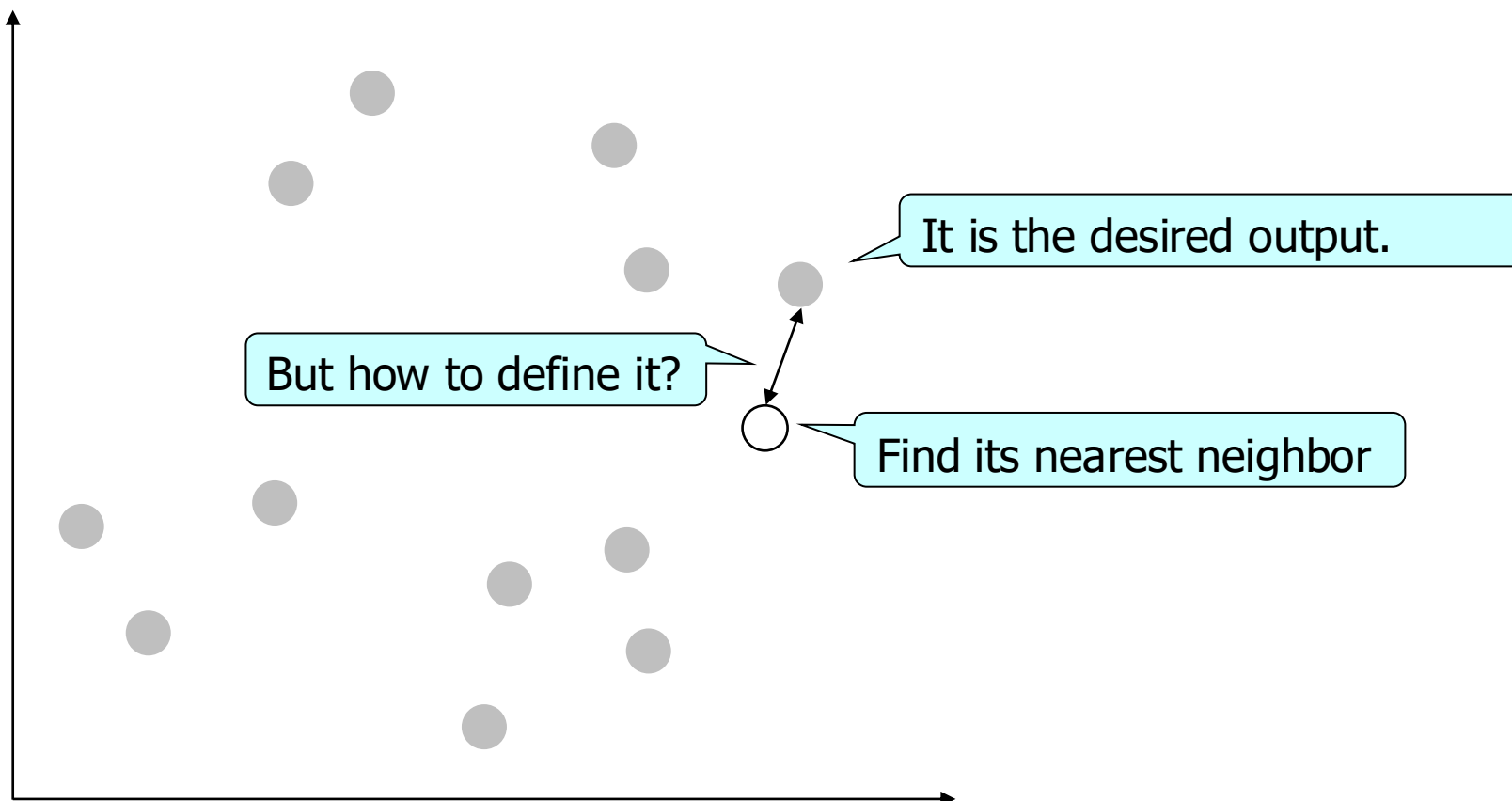
Uses NN's label as its label

Find its nearest neighbor

Task

Querying

Also called searching



Forecasting

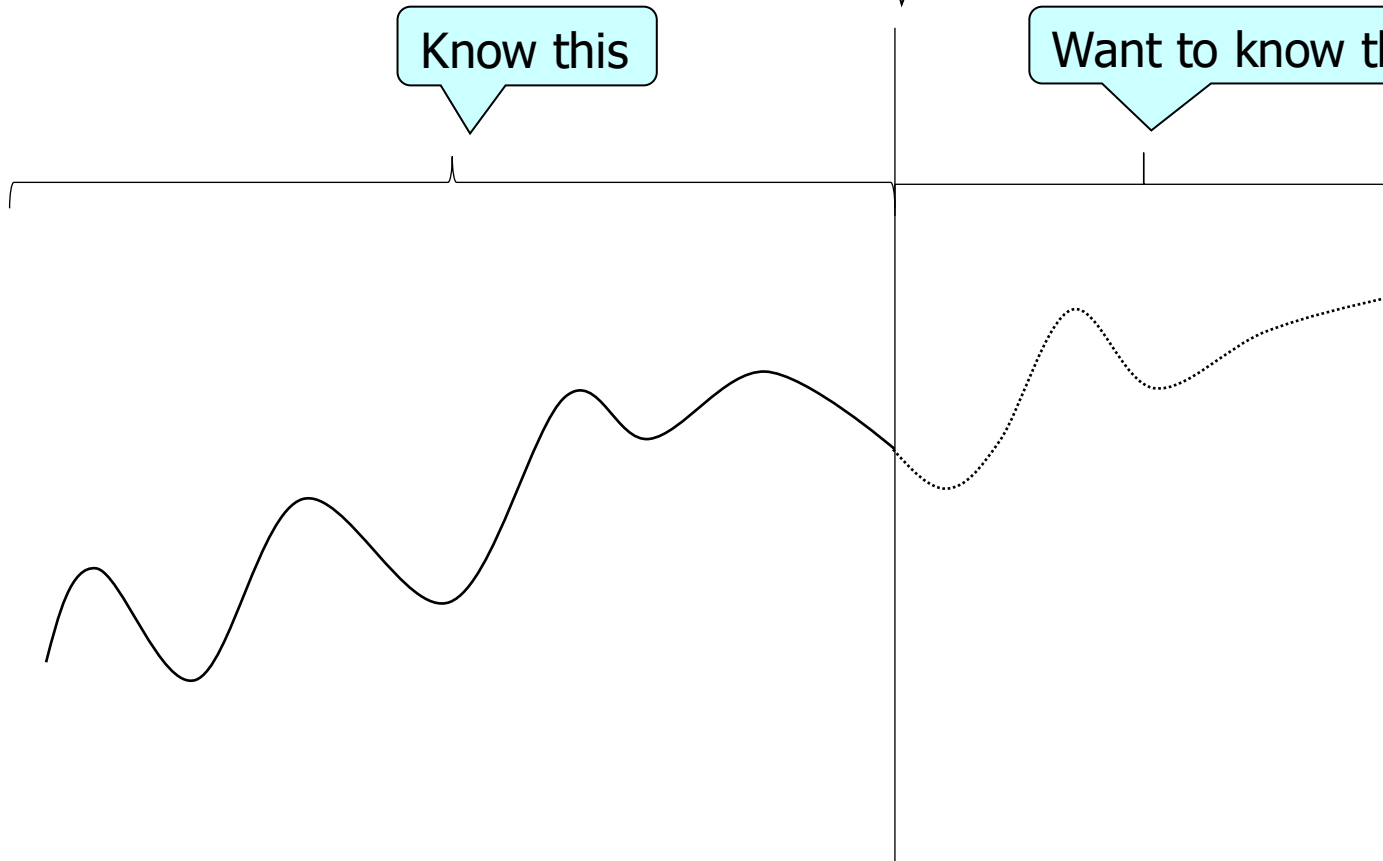
Task

Train/Test split

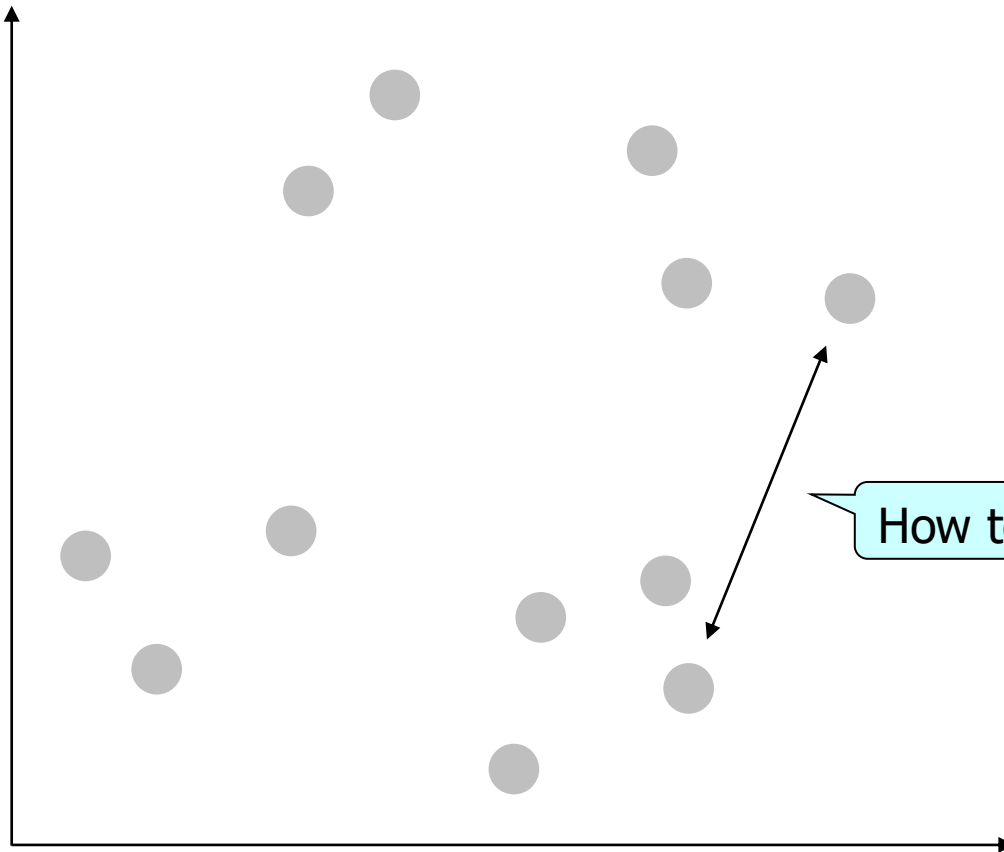
Now!

Know this

Want to know this



Similarity



Invariance

What aspect to focus/ignore

Should match our intuition

How to define it?

Distance measure (Similarity)

In different applications, different invariance

Invariance in Images

Input



Rotation



Mirror



Size



Not quite?

Sometimes you can see people use a mirror-image transformation on YouTube videos to bypass the copyright detector.

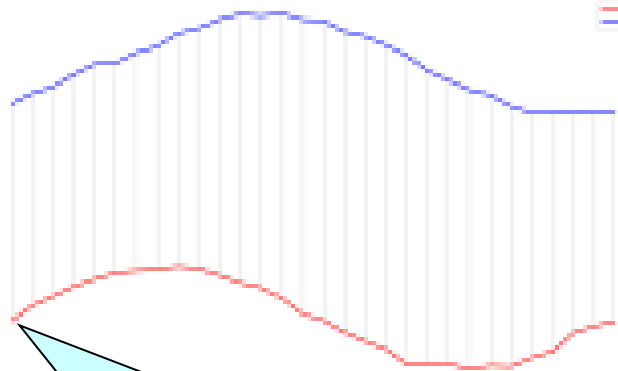
The most common similarity measure in Time Series

Euclidean distance & Z-normalization

just offset

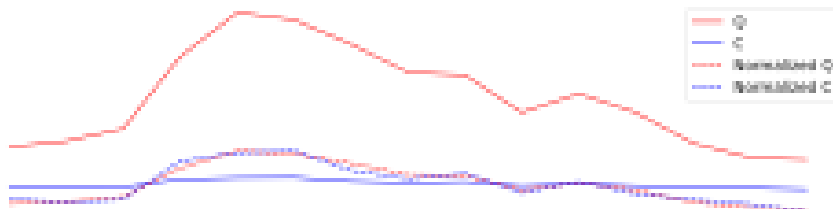
Definition Euclidean Distance (ED)). Given two series Q and C both with length n , the Euclidean Distance between them is defined as:

$$ED(Q, C) = \sqrt{\sum_{i=1}^n (q_i - c_i)^2} \quad (1)$$



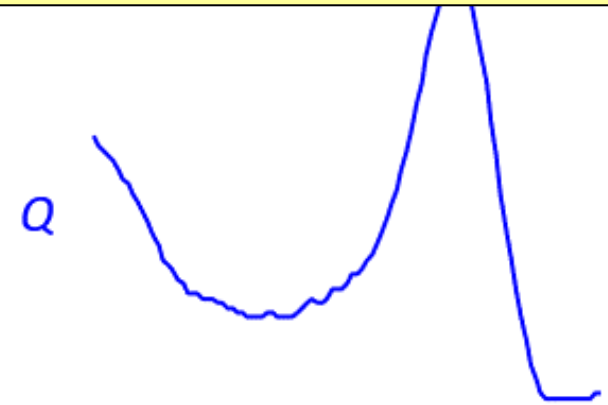
1st point to 1st point

Lock-step

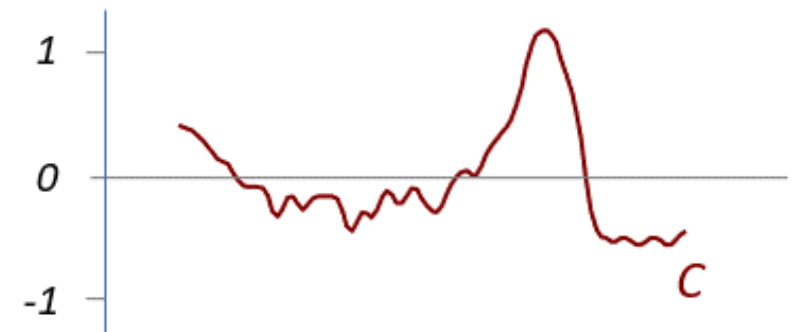


Z-normalization. Resulting time series have mean = 0 and std = 1.

$$(T - \text{mean}(T)) / \text{std}(T)$$



$D(Q, C)$



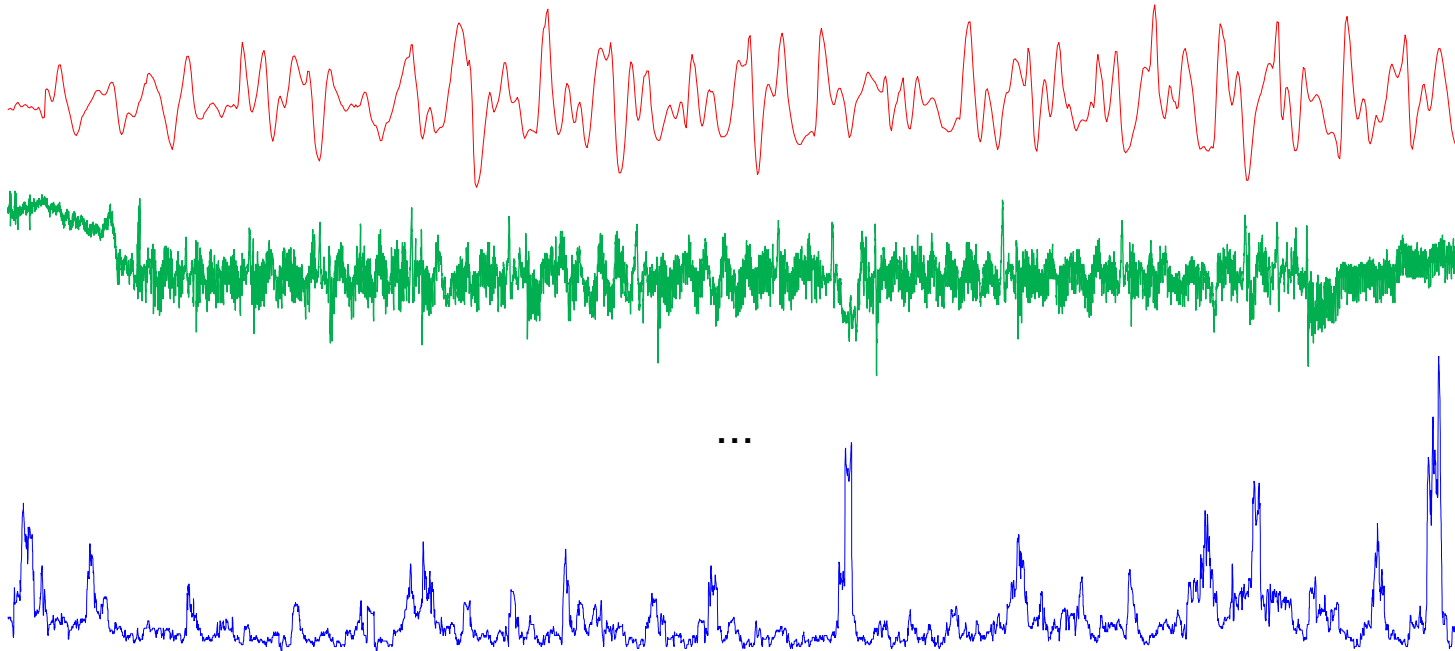
Multivariate time series

Sensor 1

Sensor 2

...

Sensor C



Otherwise,
padding

Need to have the
same length

A Channel

Each entry is a
vector of length C

Or you can think there are
C parallel time series



Outline

We also have some evaluation metrics to determine "goodness".

It means we have data and machine learning models

It is a bioinformatics study, tackled by a machine learning way.

- Introduction
- 1. MTSCCleav: a Multivariate Time Series Classification (MTSC)-based Method for Predicting Human Dicer Cleavage
- 2. Scaling with Multiple Scaling Factors in Time Series Searching
- 3. Leveraging Nearest Neighbors for Time Series Forecasting with Matrix Profile
- Conclusion

Task



Bioinformatics

- Biology

Why is it important?

- Biotechnology

What kind of data can we observe?

- Informatics

- Prepare/Curate the dataset

Define input/output relationship

- Augment the dataset (Secondary Structure)

Covariates

9 encoding methods

- Transformation into a Multivariate time series

Find useful information

- Convolution-Based Classifiers

- ROCKETS and its variants

5 classifiers

Use convolution to extract some information

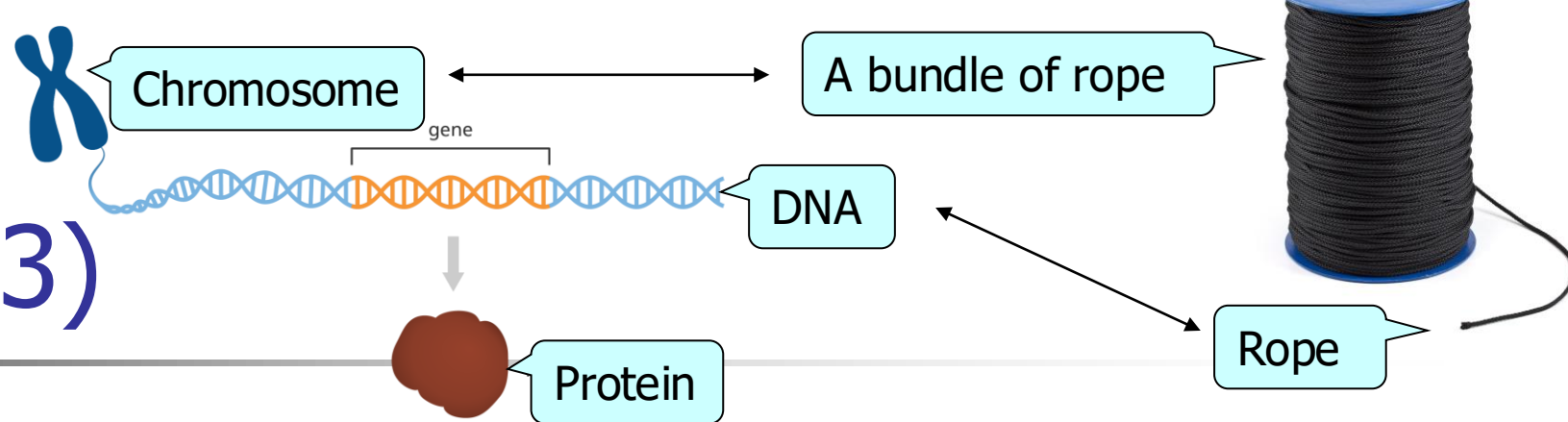
This information is deemed to capture some high-level concept

Then feed into a simple classifier. A ridge classifier is used.

The location is a covariate

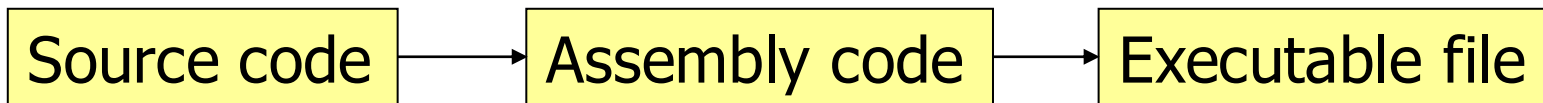
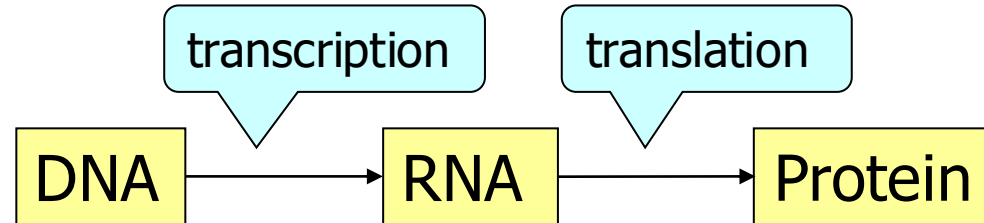
For example, if a teenager lives near the campus, he is more likely to be a college student.

Biology (1/3)

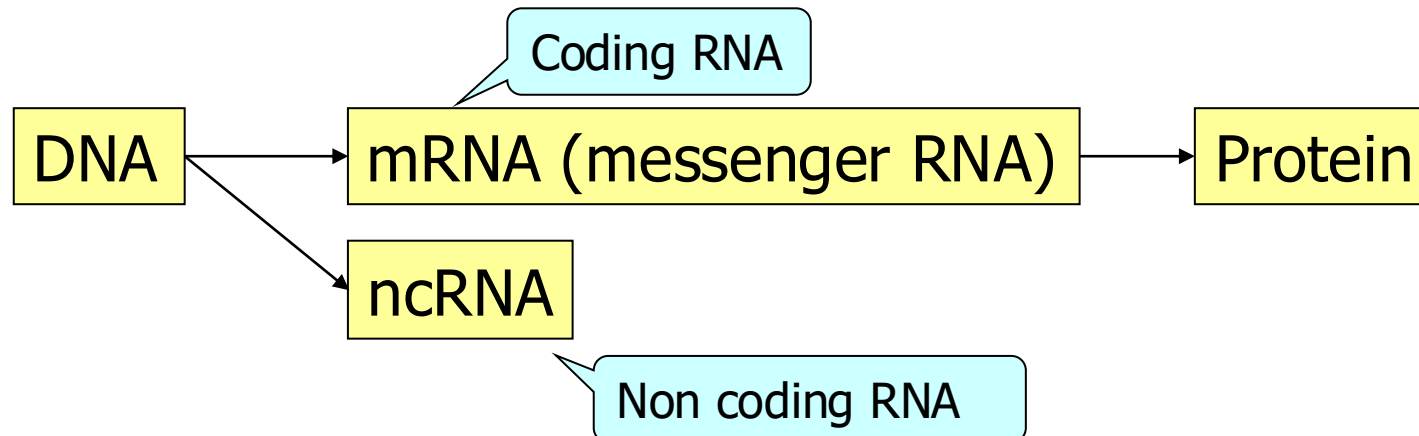


- Central dogma = Flow of genetic information

General



In detail



Biology (2/3)

■ ncRNA

- siRNA (small interfering RNA)
- miRNA (microRNA)

The 2006 Nobel Prize in Physiology or Medicine

Important discovery

RNA interference (RNAi)

Shut down a gene by degrading the corresponding mRNA

The generation (biosynthesis) of miRNA is important

Phase 1

cuts

■ miRNA

Dorsal & DCGR8 cleaves pri-miRNA and forms pre-miRNA

- primary miRNA (pri-miRNA) → precursor miRNA (pre-miRNA)

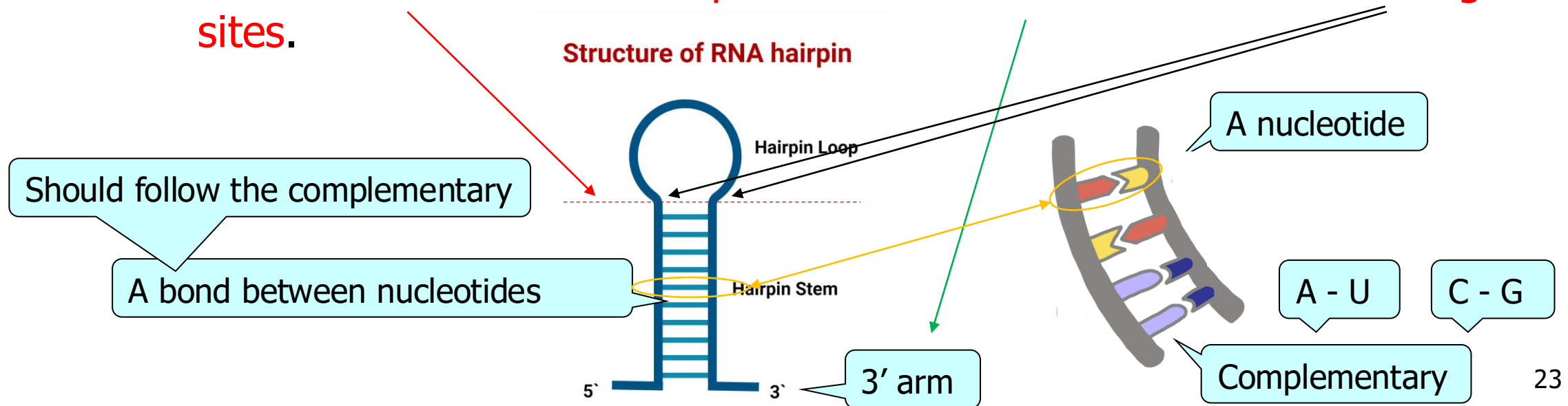
Phase 2

An enzyme called Dicer cleaves pri-miRNA and forms miRNA

- pre-miRNA → miRNA

Biology (3/3)

- A pri-miRNA contains a hairpin loop
- Phase 1: The loop is still preserved
- Phase 2 (Dicer): The loop is gone
 - Dicer cleaves the **stem-loop** from the two **arms** at the two **cleavage sites**.



Biotechnology

How we see/measure the data

- miRbase database

Accession	Name	Organism	Sequence	Mature miRNA 1	Mature miRNA 2
MI0000001	cel-let-7	Caenorhabditis elegans	<i>UACAC ... UUGGA</i>	cel-let-7-3p 17:38 experimental	cel-let-7-3p 60:81 experimental
MI0000060	hsa-let-7a-1	Homo sapiens	<i>UGGGA ... UCCUA</i>	hsa-let-7a-3p 6:27 experimental	hsa-let-7a-3p 57:77 experimental
MI0000114	hsa-mir-107	Homo sapiens	<i>CUCUC ... ACAGA</i>	hsa-miR-107 50:72 experimental	NA
MI0000238	hsa-mir-196a-1	Homo sapiens	<i>GUGAA ... UUCAC</i>	hsa-miR-196a-5p 7:28 experimental	hsa-miR-196a-1-3p 45:65 not experimental

positions

- But wait. Why is it 1D? Biological molecules should be 3D!
 - Because the methods of “seeing” 3D are expensive
 - E.g. X-ray crystallography or nuclear magnetic resonance (NMR)

- We use some tools to recover the partial structure
 - Secondary Structure

Sequence	Secondary Structure (In Dot-bracket notation)
1 UGGGAUGAGGUAGUAGGUUGUAUAGUU 27	1 (((((.....))) 27
28 UUAGGGUACACCCACCACUGGGAGAU 54	28 UUAGGGUACACCCACCACUGGGAGAU 54
55 AAUAUACAUAUACUGUCUUUCCUA 80	55))))))) 80
Base-pair probabilities sequence (the first 10 bases)	
1 (0.549, 0.946, 0.987, 0.987, 0.904) 5	
6 (0.000, 0.841, 0.974, 0.981, 0.890) 10	

pri-miRNA

Similarly, 3p (non-) cleavage pattern

4 patterns
= 4 strings

Directionality


Color intensity:
Probability

5p non-cleavage pattern

A pattern that is 6 nt (nucleotide) away

Length = 14

Cleavage site



5p cleavage pattern

A pattern that having cleavage site at the center

Length = 14

34

28

21

What do we have now?

Convert to time series

1

2

	5p cleav	5p non-cleav	3p cleav	3p non-cleav
Input strand	UAUAGUUUAAGGGU	AGGUUGUAUAGUUU	GAGAUAACUAUACA	ACUAUACAAUCUAC
Complementary strand	AUAUCAA_____UA	UCUAACAUUAUCAA_	C_CUGUUGAUUAUGU	UGAUAUGUUGGAUG

But what we also have:

"(" maps to 1, "." maps to 0, and ")" maps to -1

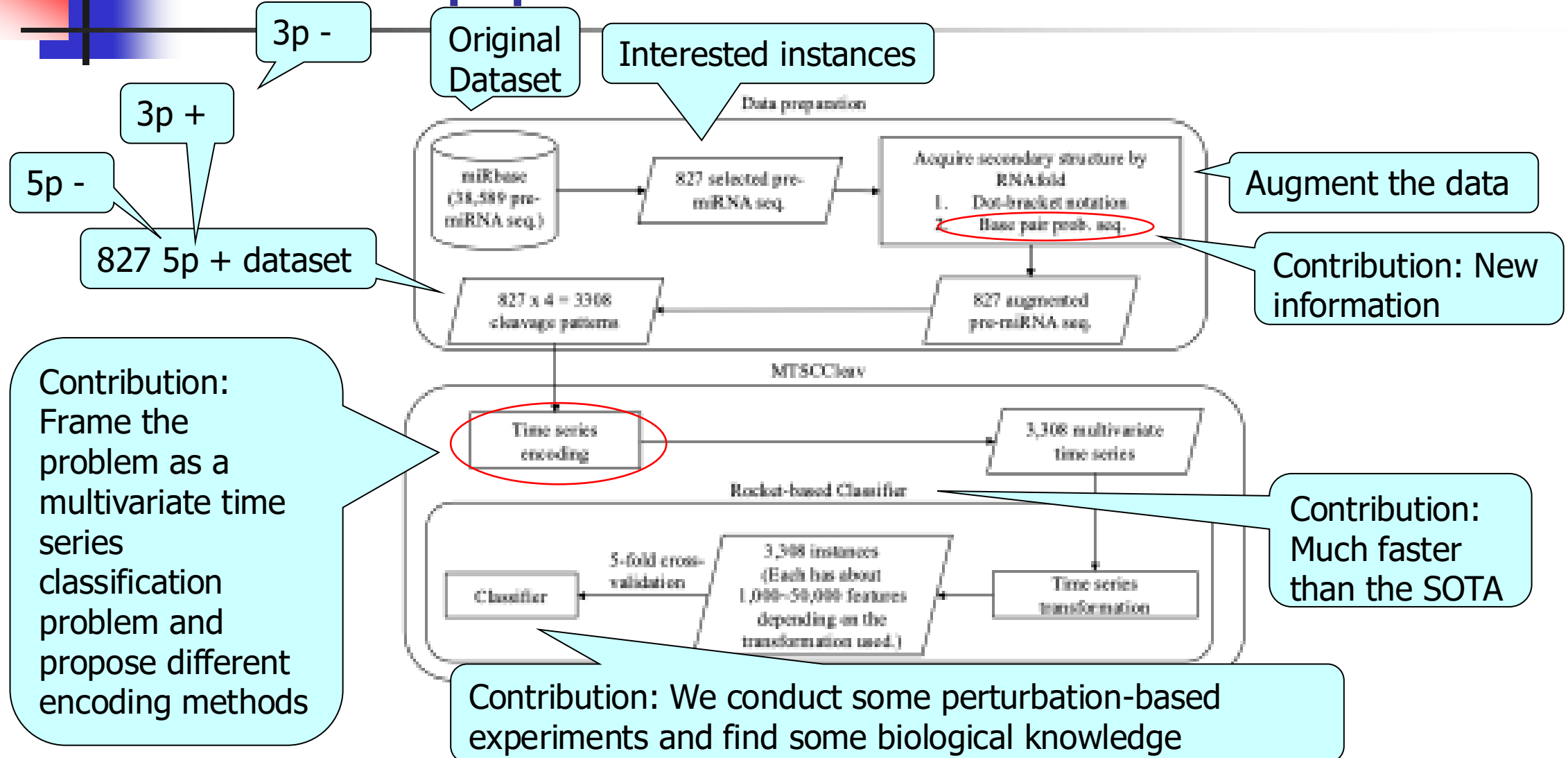
3 Secondary Structure between the two strands (in dot-bracket notation)

4 Probability sequence of each strand

Already time series

Two Binary dataset and one multi-class dataset

Overall pipeline



Confidence/ weight

Probability sequence

Example

Transformation (1/4)

$S = C, _, C, U, G, U, U, G, A, U$
 $P = 0.843, 0.000, 0.807, 0.807, 0.793,$
 $0.914, 0.982, 1.000, 0.999, 0.999$

Single value mapping

Encode a string into a time series

Because of the domain knowledge

```
for i = 1 to |S|:  
   $x_i = \begin{cases} 2 \cdot p_1 & \text{if } s_i = A \\ 1 \cdot p_1 & \text{if } s_i = G \\ -1 \cdot p_1 & \text{if } s_i = C \\ -2 \cdot p_1 & \text{if } s_i = U \\ 0 & \text{otherwise} \end{cases}$   
return T
```

Without base-pair probability sequence:
 $T = -1, 0, -1, -2, 1, -2, -2, 1, 2, -2$

With base-pair probability sequence:
 $T = -0.843, 0.000, -0.807, -1.614,$
 $0.793, -1.829, -1.963,$
 $1.000, 1.999, -1.998$

Encode a string into two time series with different lengths

Grouped variable-length channel mapping

```
j = 1, k = 1  
for i = 1 to |S|:  
   $x_j^1 = \begin{cases} 1 \cdot p_1 & \text{if } s_i = A \\ -1 \cdot p_1 & \text{if } s_i = U \\ 0 & \text{otherwise} \end{cases}$   
   $x_k^2 = \begin{cases} 1 \cdot p_1 & \text{if } s_i = G \\ -1 \cdot p_1 & \text{if } s_i = C \end{cases}$   
  if  $(s_i = G) \text{ or } (s_i = C)$ :  
    increment k by 1  
  else:  
    increment j by 1  
return  $T^1, T^2$ 
```

Without base-pair probability sequence:
 $T^1 = 0, -1, -1, -1, 1, -1$
 $T^2 = -1, -1, 1, 1$

With base-pair probability sequence:
 $T^1 = 0.000, -0.807, -0.914, -0.982, 0.999, -0.999$
 $T^2 = -0.843, -0.807, 0.793, 1.000$

Transformation (2/4)

- Grouped fixed-length channel mapping

Encode a string into two time series with the same lengths

```
for i = 1 to |S|:  
     $r_i^1 = \begin{cases} 1 \cdot p_A & \text{if } s_i = A \\ -1 \cdot p_U & \text{if } s_i = U \\ 0 & \text{otherwise} \end{cases}$   
     $r_i^2 = \begin{cases} 1 \cdot p_G & \text{if } s_i = G \\ -1 \cdot p_C & \text{if } s_i = C \\ 0 & \text{otherwise} \end{cases}$   
return  $T^1, T^2$ 
```

Without base-pair probability sequence:

$T^1 = 0, 0, 0, -1, 0, -1, -1, 0, 1, -1$

$T^2 = -1, 0, -1, 0, 1, 0, 0, 1, 0, 0$

With base-pair probability sequence:

$T^1 = 0.000, 0.000, 0.000, -0.807,$
 $0.000, -0.914, -0.982,$
 $0.000, 0.999, -0.9999$
 $T^2 = -0.843, 0.000, -0.807, 0.000,$
 $0.793, 0.000, 0.000,$
 $1.000, 0.000, 0.000$



Transformation (3/4)

- Or we can use the cumulative approach instead of the absolute approach

- Cumulative mapping

```

$$A_1 = 0$$

$$\text{for } i = 1 \text{ to } |S|:$$

$$A_{i+1} = \begin{cases} A_i + 2 \cdot p_i & \text{if } x_i = A \\ A_i + 1 \cdot p_i & \text{if } x_i = G \\ A_i - 1 \cdot p_i & \text{if } x_i = C \\ A_i - 2 \cdot p_i & \text{if } x_i = U \\ A_i & \text{otherwise} \end{cases}$$

$$\text{return } T \text{ s.t. } |T| = |S| + 1$$

```

- Cumulative grouped variable-length channel mapping

- Cumulative grouped fixed-length channel mapping

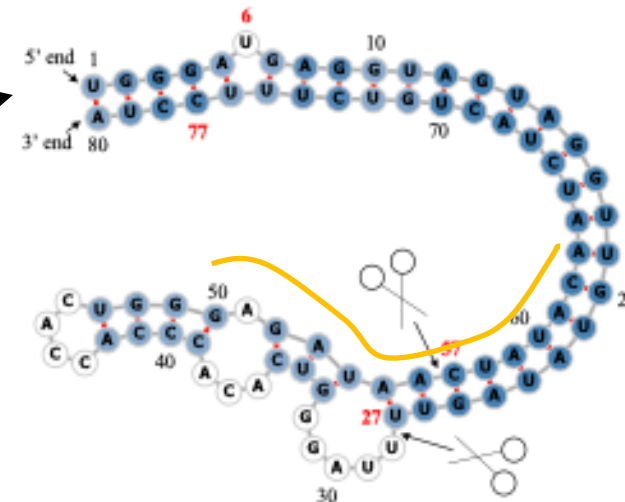
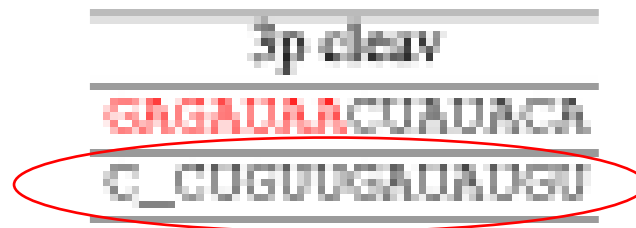
Cumulative

- In the cumulative method, we can choose where to start the cumulation

- From the start of the pattern

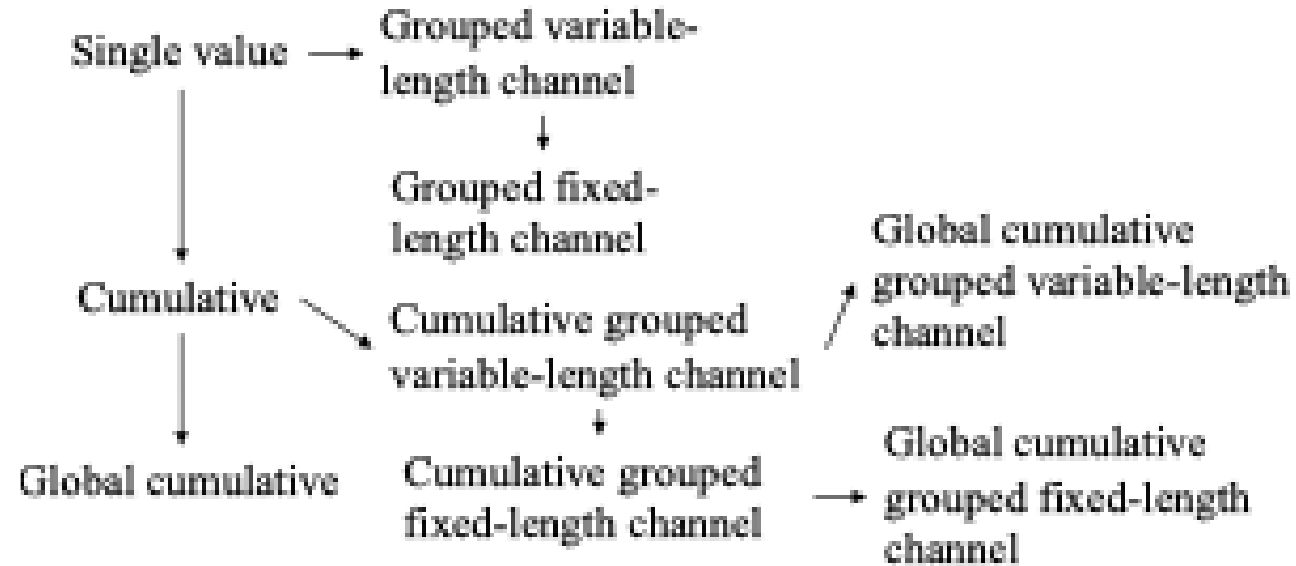
Example
 $S = C, _, C, U, G, U, U, G, A, U$
 $P = 0.843, 0.000, 0.807, 0.807, 0.793, 0.914, 0.982, 1.000, 0.990, 0.990$

- Or from the start of the origin





Summary of 9 Encoding methods



How many channels of time series do we have now

- Three types of data (the input features for each instance)
 - RNA sequence
 - Primary strand
 - Complementary strand
 - Secondary structure information
 - Base pair probability sequence

"(" maps to 1
"." maps to 0
")" maps to -1

A time series

Performance across the channel

Secondary structure is a good covariate

Standalone is better.

Classifier		Sp arm					Sp arm					multi-class				
		Acc	Sp	Sa	F1	MCC	Acc	Sp	Sa	F1	MCC	Acc	Sp	Sa	F1	MCC
Baseline (cfg 1)	ROCKET	0.781	0.743	0.819	0.789	0.563	0.790	0.773	0.807	0.793	0.580	0.717	0.838	0.685	0.700	0.538
	MiniROCKET	0.755	0.728	0.782	0.762	0.512	0.788	0.781	0.794	0.789	0.576	0.685	0.823	0.653	0.662	0.486
	MultiROCKET	0.784	0.767	0.801	0.787	0.569	0.803	0.792	0.814	0.805	0.606	0.691	0.830	0.667	0.672	0.501
	Hydra	0.830	0.800	0.860	0.835	0.663	0.808	0.797	0.820	0.810	0.617	0.731	0.844	0.696	0.712	0.560
	MultiROCKET-Hydra	0.796	0.778	0.815	0.800	0.594	0.807	0.767	0.816	0.808	0.614	0.701	0.836	0.681	0.686	0.520
Baseline + Secondary Structure (cfg 2)	ROCKET	0.847	0.832	0.862	0.849	0.695	0.855	0.842	0.868	0.857	0.711	0.836	0.907	0.828	0.833	0.736
	MiniROCKET	0.825	0.807	0.843	0.827	0.652	0.822	0.802	0.843	0.826	0.646	0.823	0.900	0.812	0.818	0.715
	MultiROCKET	0.812	0.800	0.822	0.814	0.626	0.824	0.809	0.830	0.826	0.649	0.796	0.888	0.791	0.792	0.673
	Hydra	0.845	0.816	0.873	0.849	0.691	0.846	0.817	0.874	0.850	0.693	0.830	0.901	0.814	0.826	0.724
	MultiROCKET-Hydra	0.817	0.809	0.826	0.819	0.635	0.825	0.816	0.834	0.826	0.652	0.803	0.891	0.798	0.800	0.684
Baseline + Base-pair probability (Standalone) (cfg 3)	ROCKET	0.842	0.828	0.855	0.844	0.684	0.855	0.856	0.854	0.855	0.710	0.795	0.885	0.783	0.789	0.670
	MiniROCKET	0.817	0.820	0.814	0.816	0.634	0.836	0.834	0.838	0.836	0.673	0.772	0.872	0.757	0.764	0.632
	MultiROCKET	0.822	0.813	0.832	0.824	0.645	0.825	0.831	0.820	0.824	0.651	0.758	0.866	0.747	0.750	0.612
	Hydra	0.846	0.827	0.865	0.849	0.693	0.851	0.840	0.861	0.852	0.702	0.789	0.879	0.769	0.780	0.658
	MultiROCKET-Hydra	0.822	0.809	0.834	0.824	0.644	0.835	0.840	0.830	0.834	0.670	0.759	0.866	0.746	0.750	0.611
Baseline + Base-pair probability (Incorporated) (cfg 4)	ROCKET	0.799	0.771	0.827	0.805	0.600	0.809	0.786	0.832	0.813	0.619	0.737	0.850	0.712	0.724	0.573
	MiniROCKET	0.776	0.756	0.797	0.781	0.554	0.801	0.806	0.794	0.799	0.603	0.705	0.835	0.675	0.684	0.521
	MultiROCKET	0.814	0.801	0.828	0.817	0.630	0.816	0.812	0.820	0.816	0.634	0.726	0.848	0.706	0.712	0.556
	Hydra	0.822	0.787	0.857	0.828	0.647	0.834	0.828	0.840	0.835	0.669	0.759	0.862	0.734	0.746	0.608
	MultiROCKET-Hydra	0.814	0.802	0.820	0.817	0.629	0.820	0.825	0.816	0.819	0.642	0.736	0.853	0.717	0.723	0.574

Primary strand: 1 or 2 channels

Complementary strand: 1 or 2 channels

Secondary structure: 1 channel

Probability sequence: 1 channel

Classifier

A time series

Activation Map

A kernel

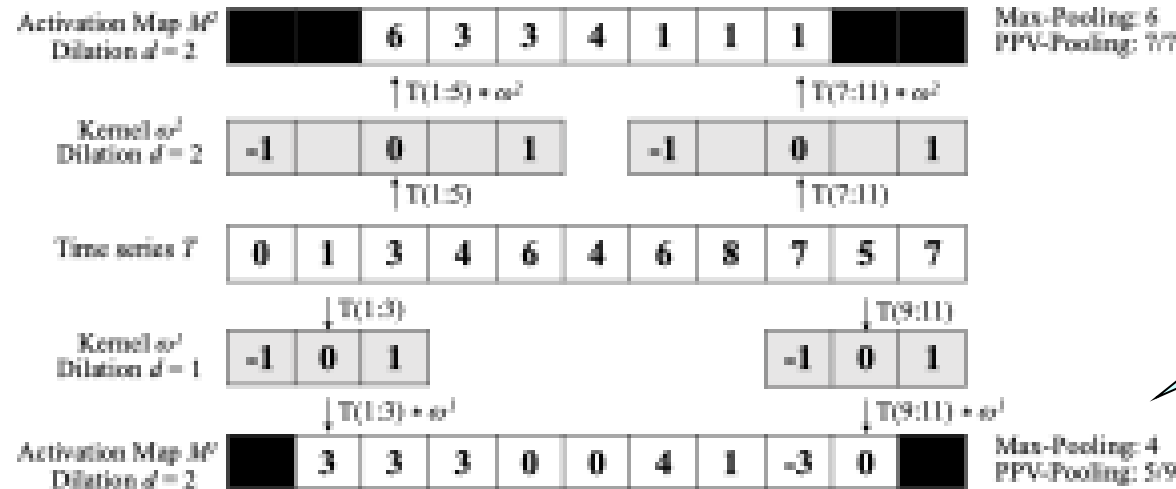
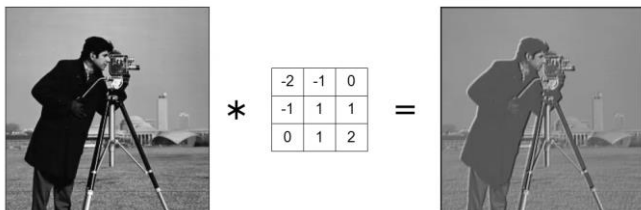
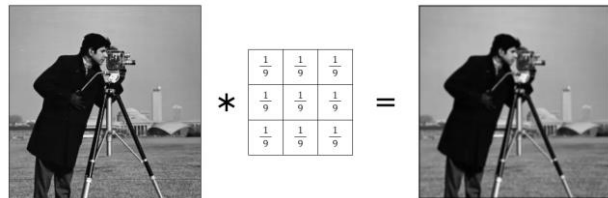
A time series

Sliding window


$$M_i = T(i:i+l-1) * \omega = \sum_{j=0}^{l-1} t_{i+j} \cdot \omega_{1+j}$$

■ Convolution-Based Classifiers

- It uses Convolution to create a time series M of a given time series T
- Use M's statistics as features.
- Then feed those statistics into other classifier such as Ridge classifier



Comparison of rocket-based classifiers

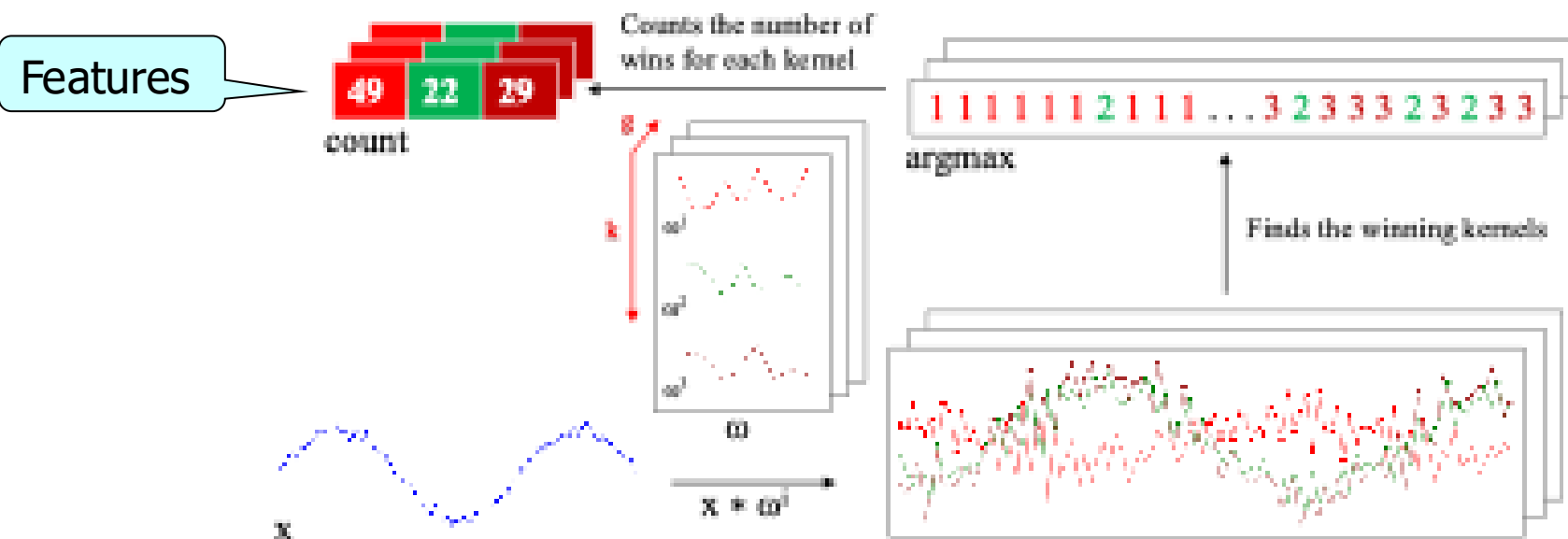


	ROCKET	MiniROCKET	MultiROCKET	Hydra
kernel length	{7, 9, 11}	9	9	9
kernel weights	$\mathcal{N}(0, 1)$	$\{-1, 2\}$	$\{-1, 2\}$	$\mathcal{N}(0, 1)$
bias	$\mathcal{U}(0, 1)$	from output	from output	none
dilation	random	fixed (input-relative)	fixed (input-relative)	random
padding	random	fixed	fixed	always
pooling operations	MAX, PPV	PPV	PPV, MPV, MIPV, LSPV	Response per Kernel/Group
1 st order difference	no	no	yes	yes
feature vector size	20k	10k	50k	relative to input

MultiROCKET-Hydra

- $\mathcal{N}(0, 1)$: a standard normal distribution
- $\mathcal{U}(0, 1)$: a uniform distribution

Hydra



The features are then going into

Ridge regression

Used as a classifier

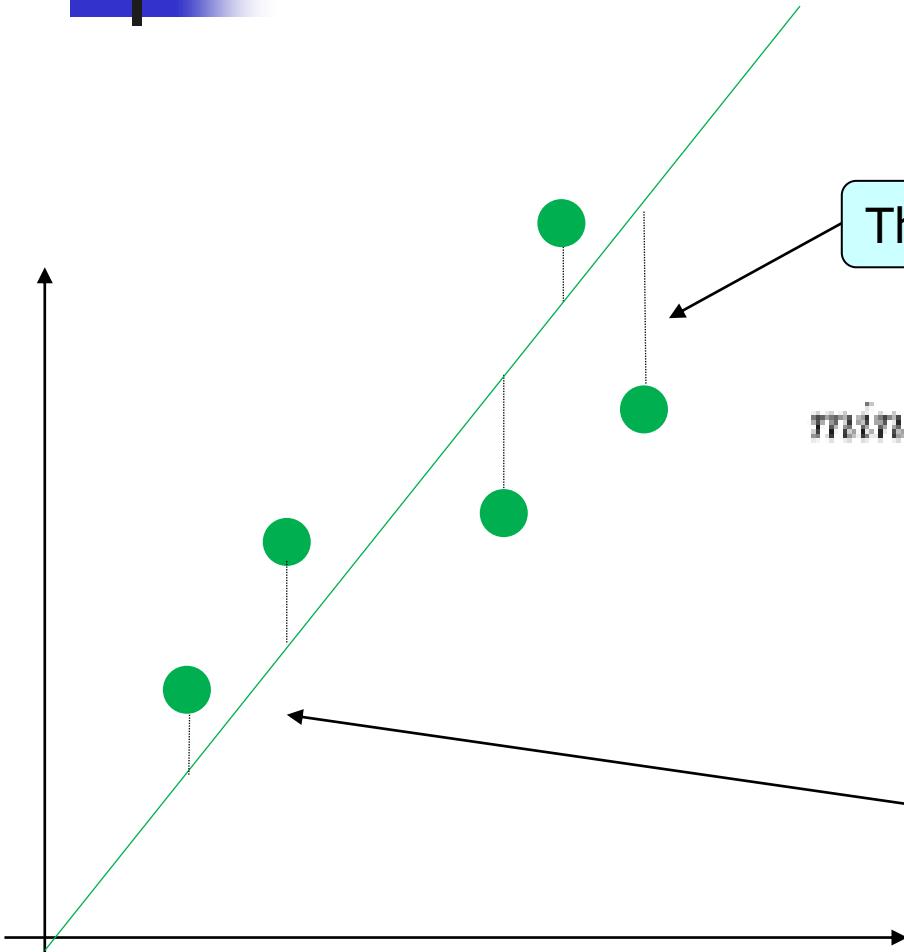
The sum of the residual

$$\min_w \left(\sum_{i=1}^n (x_i^T w - y_i)^2 + \lambda \|w\|_2^2 \right)$$

L₂ norm of the weight vector

L₂ penalty: Used to introduce bias in the fit to avoid overfitting

When the slope is small, y-value is less sensitive to changes in x-value





Evaluation Metric

$$Acc = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Sp = \frac{TN}{TN + FP}$$

$$Sn = \frac{TP}{TP + FN}$$

$$F1 = \frac{2 \times TP}{2 \times TP + FP + FN}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

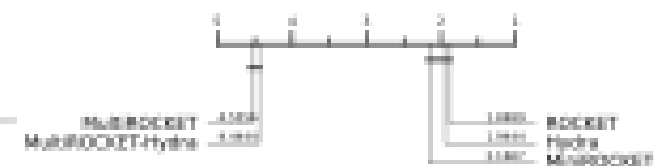
$$MCC_{multi} = \frac{c \times s - \sum_k^K p_k \times t_k}{\sqrt{(s^2 - \sum_k^K p_k^2) \times (s^2 - \sum_k^K t_k^2)}}$$

Think of it as
different
datasets

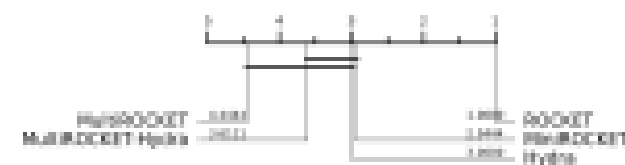
Results

CD diagrams

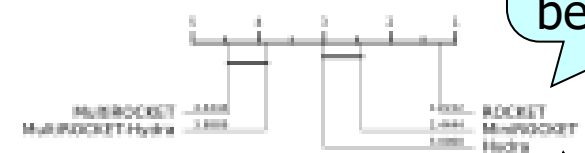
	Classifier	Sp arm					Sp arm					multi-class				
		Acc	Sp	Sn	F1	MCC	Acc	Sp	Sn	F1	MCC	Acc	Sp	Sn	F1	MCC
Single value mapping (enc 1)	ROCKET	0.849	0.842	0.857	0.851	0.609	0.863	0.854	0.875	0.865	0.727	0.853	0.917	0.847	0.851	0.764
	MiniROCKET	0.823	0.809	0.837	0.825	0.647	0.823	0.828	0.817	0.822	0.647	0.835	0.906	0.828	0.833	0.735
	MultiROCKET	0.821	0.802	0.840	0.824	0.643	0.839	0.826	0.852	0.841	0.679	0.811	0.894	0.806	0.806	0.697
	Hydra	0.843	0.820	0.867	0.847	0.688	0.838	0.819	0.857	0.841	0.677	0.831	0.901	0.815	0.827	0.727
	MultiROCKET-Hydra	0.820	0.803	0.837	0.823	0.640	0.840	0.830	0.850	0.841	0.680	0.816	0.896	0.810	0.814	0.704
Grouped variable-length channel mapping (enc 2)	ROCKET	0.835	0.826	0.844	0.836	0.670	0.855	0.849	0.861	0.856	0.710	0.846	0.913	0.839	0.844	0.752
	MiniROCKET	0.843	0.833	0.853	0.844	0.686	0.831	0.821	0.842	0.833	0.663	0.837	0.907	0.828	0.834	0.757
	MultiROCKET	0.819	0.809	0.828	0.820	0.638	0.817	0.814	0.820	0.818	0.634	0.890	0.894	0.806	0.808	0.695
	Hydra	0.825	0.780	0.869	0.852	0.653	0.811	0.769	0.854	0.819	0.626	0.818	0.892	0.765	0.812	0.705
	MultiROCKET-Hydra	0.818	0.814	0.822	0.819	0.636	0.831	0.825	0.837	0.832	0.662	0.820	0.900	0.815	0.818	0.710
Grouped fixed-length channel mapping (enc 3)	ROCKET	0.851	0.843	0.859	0.852	0.702	0.863	0.850	0.875	0.864	0.726	0.849	0.915	0.843	0.847	0.757
	MiniROCKET	0.844	0.836	0.853	0.845	0.689	0.840	0.826	0.855	0.843	0.682	0.851	0.915	0.844	0.849	0.760
	MultiROCKET	0.811	0.815	0.848	0.834	0.663	0.824	0.813	0.836	0.826	0.649	0.811	0.896	0.808	0.808	0.698
	Hydra	0.848	0.816	0.880	0.853	0.699	0.862	0.839	0.884	0.864	0.724	0.843	0.908	0.837	0.839	0.746
	MultiROCKET-Hydra	0.836	0.813	0.859	0.839	0.672	0.833	0.820	0.845	0.835	0.665	0.828	0.905	0.824	0.826	0.725
Cumulative mapping (enc 4)	ROCKET	0.850	0.834	0.866	0.852	0.701	0.863	0.855	0.871	0.864	0.726	0.852	0.915	0.842	0.850	0.762
	MiniROCKET	0.840	0.821	0.860	0.843	0.682	0.840	0.821	0.844	0.841	0.682	0.843	0.911	0.835	0.840	0.747
	MultiROCKET	0.822	0.809	0.834	0.824	0.644	0.832	0.830	0.834	0.832	0.665	0.820	0.898	0.810	0.816	0.709
	Hydra	0.848	0.819	0.878	0.853	0.698	0.853	0.856	0.869	0.855	0.705	0.845	0.910	0.830	0.841	0.749
	MultiROCKET-Hydra	0.824	0.811	0.856	0.825	0.647	0.838	0.833	0.843	0.839	0.677	0.821	0.898	0.810	0.817	0.711
Cumulative grouped variable-length channel mapping (enc 5)	ROCKET	0.843	0.821	0.866	0.847	0.688	0.856	0.840	0.871	0.857	0.712	0.855	0.916	0.843	0.851	0.766
	MiniROCKET	0.845	0.826	0.865	0.848	0.691	0.836	0.833	0.838	0.836	0.672	0.840	0.909	0.833	0.838	0.742
	MultiROCKET	0.826	0.814	0.838	0.828	0.653	0.815	0.820	0.810	0.814	0.631	0.826	0.902	0.820	0.824	0.721
	Hydra	0.850	0.819	0.880	0.854	0.701	0.834	0.807	0.861	0.838	0.669	0.833	0.903	0.818	0.829	0.731
	MultiROCKET-Hydra	0.824	0.810	0.838	0.826	0.649	0.833	0.833	0.833	0.833	0.666	0.830	0.903	0.821	0.827	0.726
Cumulative grouped fixed-length channel mapping (enc 6)	ROCKET	0.856	0.836	0.876	0.858	0.712	0.870	0.861	0.879	0.871	0.741	0.863	0.921	0.852	0.860	0.780
	MiniROCKET	0.856	0.837	0.874	0.858	0.712	0.842	0.839	0.845	0.843	0.685	0.845	0.912	0.837	0.843	0.751
	MultiROCKET	0.820	0.802	0.839	0.824	0.642	0.798	0.798	0.798	0.798	0.597	0.809	0.894	0.806	0.807	0.694
	Hydra	0.850	0.814	0.885	0.855	0.701	0.855	0.840	0.869	0.857	0.711	0.847	0.910	0.831	0.843	0.752
	MultiROCKET-Hydra	0.820	0.801	0.839	0.823	0.641	0.807	0.813	0.802	0.806	0.615	0.821	0.900	0.817	0.819	0.713
Global Cumulative mapping (enc 7)	ROCKET	0.850	0.834	0.866	0.852	0.701	0.863	0.855	0.871	0.864	0.726	0.852	0.915	0.842	0.850	0.762
	MiniROCKET	0.847	0.832	0.862	0.849	0.695	0.848	0.839	0.857	0.850	0.697	0.845	0.911	0.836	0.843	0.750
	MultiROCKET	0.827	0.819	0.834	0.828	0.653	0.847	0.842	0.853	0.848	0.695	0.825	0.901	0.817	0.822	0.718
	Hydra	0.851	0.821	0.880	0.855	0.709	0.861	0.848	0.874	0.863	0.722	0.847	0.911	0.834	0.844	0.753
	MultiROCKET-Hydra	0.829	0.823	0.834	0.830	0.658	0.843	0.838	0.849	0.844	0.688	0.832	0.905	0.823	0.829	0.730
Global Cumulative grouped variable-length channel mapping (enc 8)	ROCKET	0.840	0.814	0.867	0.844	0.682	0.853	0.838	0.867	0.854	0.706	0.856	0.917	0.845	0.853	0.768
	MiniROCKET	0.848	0.834	0.862	0.850	0.697	0.841	0.824	0.859	0.844	0.683	0.844	0.911	0.856	0.842	0.748
	MultiROCKET	0.834	0.828	0.839	0.834	0.668	0.831	0.821	0.842	0.833	0.663	0.828	0.904	0.823	0.826	0.724
	Hydra	0.857	0.821	0.894	0.862	0.717	0.822	0.786	0.857	0.828	0.645	0.826	0.898	0.806	0.820	0.717
	MultiROCKET-Hydra	0.837	0.834	0.839	0.837	0.674	0.834	0.827	0.840	0.835	0.668	0.835	0.907	0.828	0.832	0.754
Global Cumulative grouped fixed-length channel mapping (enc 9)	ROCKET	0.856	0.836	0.876	0.858	0.712	0.870	0.861	0.879	0.871	0.741	0.863	0.921	0.852	0.860	0.780
	MiniROCKET	0.857	0.845	0.870	0.859	0.715	0.840	0.821	0.859	0.843	0.681	0.844	0.911	0.837	0.842	0.749
	MultiROCKET	0.829	0.825	0.833	0.830	0.658	0.820	0.816	0.823	0.820	0.640	0.819	0.900	0.816	0.817	0.710
	Hydra	0.856	0.817	0.894	0.861	0.713	0.859	0.838	0.880	0.862	0.719	0.846	0.911	0.832	0.843	0.752
	MultiROCKET-Hydra	0.829	0.824	0.834	0.830	0.658	0.822	0.825	0.819	0.821	0.644	0.827	0.904	0.823	0.824	0.722



(a) 5p arm



(b) 3p arm



(c) multi-class

ROCKET is
generally the
best

MiniROCKET
is the fastest

Comparison with SOTA

Dataset	Methods	Acc	Sp	Sn	F1	MCC	Time (s)
5p arm	enc 9 + MiniROCKET	0.857	0.845	0.870	0.859	0.715	0.787
	DiCleave	0.818	0.790	0.846	0.822	0.653	21.249
3p arm	enc 9 + ROCKET	0.870	0.861	0.879	0.871	0.741	4.311
	enc 7 + MiniROCKET	0.848	0.839	0.857	0.850	0.697	0.989
	DiCleave	0.854	0.891	0.817	0.847	0.715	15.919
multi-class	enc 9 + ROCKET	0.863	0.921	0.852	0.860	0.780	12.208
	enc 3 + MiniROCKET	0.851	0.915	0.844	0.849	0.760	4.550
	DiCleave	0.820	0.895	0.804	0.815	0.710	131.151

Here, we report the best combination

Here, we report the best combination with respect to MiniROCKET

In general, enc 7/9 is the best encoding method and ROCKET is the best classifier

Perturbation experiment

Single value

Length = 14

Channel 1

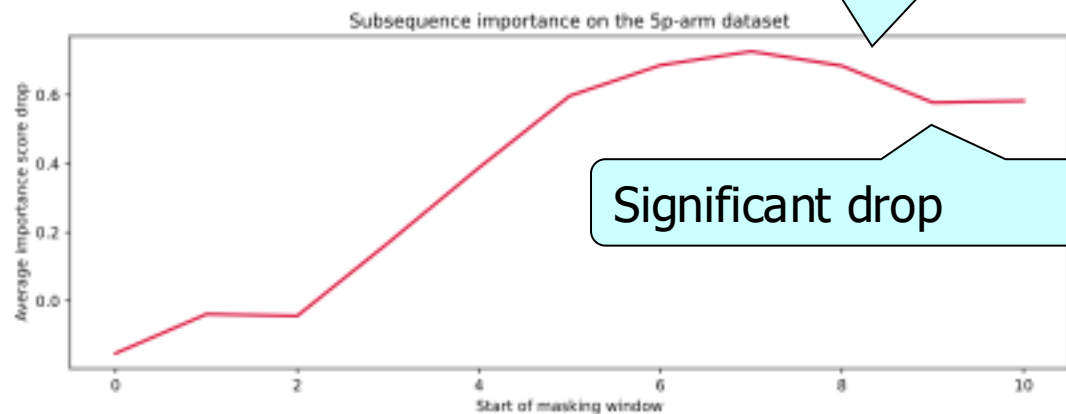
Channel 2

...

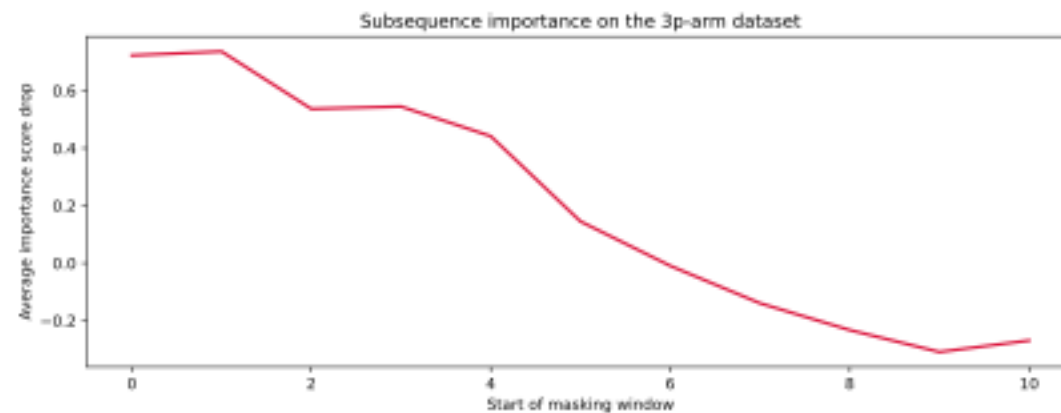
Mask of length = 4

Important

Significant drop



(a) 5p arm



(b) 3p arm



Outline

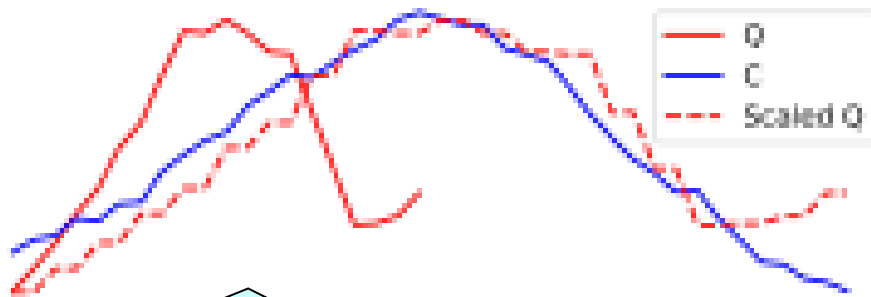
- Introduction
- 1. MTSCCleav: a Multivariate Time Series Classification (MTSC)-based Method for Predicting Human Dicer Cleavage
- **2. Scaling with Multiple Scaling Factors in Time Series Searching**
- 3. Leveraging Nearest Neighbors for Time Series Forecasting with Matrix Profile
- Conclusion

Query

Candidate

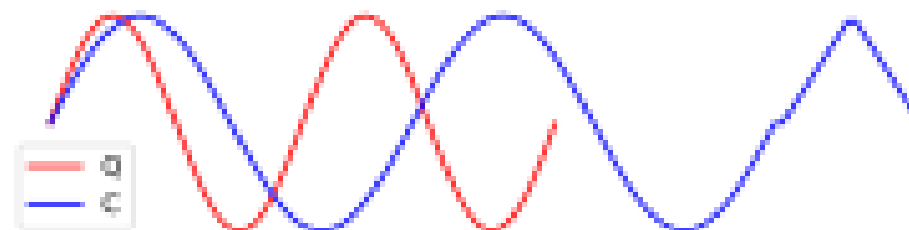
How to compare Q and C

They are in different scale



We can just scale up Q to the length of C

They are in different scale



Q is similar to a prefix of C

Before applying the distance measure, we need to eliminate the scaling effect.

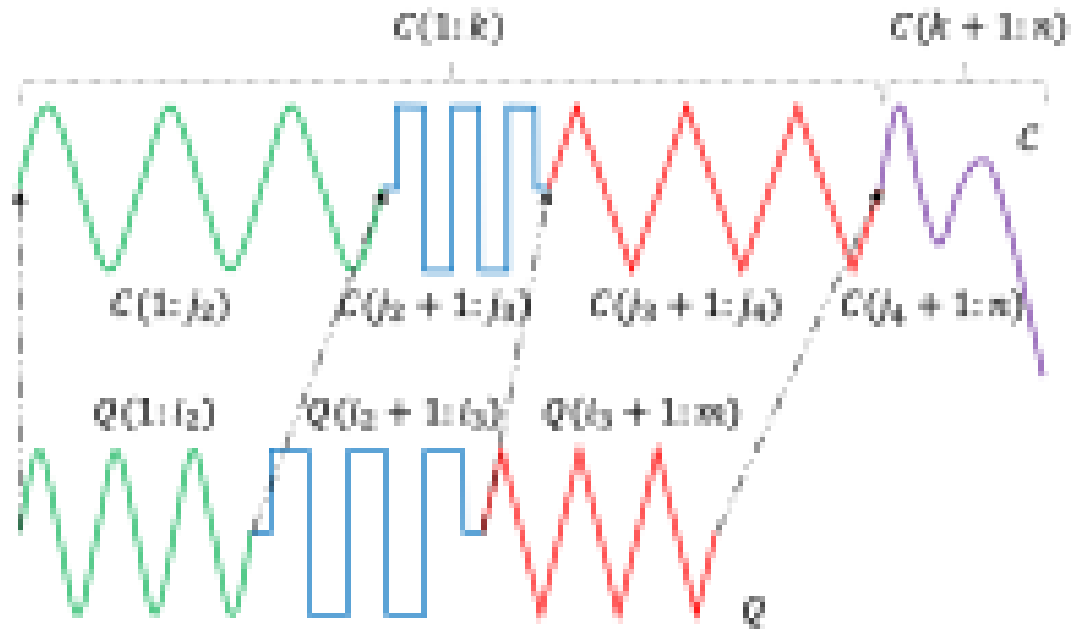
Limitation: Only one scaling factor

More than one scaling factor

Can it be solved by DTW?

But wait! What is DTW?

A famous elastic distance measure



Dynamic Time Warping (DTW)

We can speed this up by constraining the warping path. Band Constraint

$O(n^2)$

The cost matrix has a size of n^2 .

DTW is computed by filling the entries in a cost matrix

elastic

1 to many

$$DTW(Q, C) = \min_{W \in W^*} \sum_{k=1}^{|W|} w_k$$

Optimal Warping path W^*

3 pts map to 1 pt

DTW

5 pts map to 1 pt

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

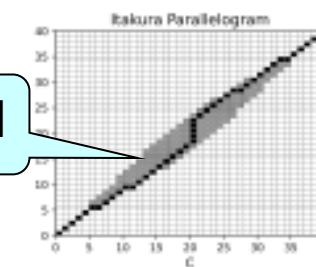
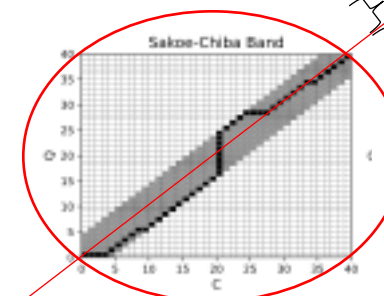
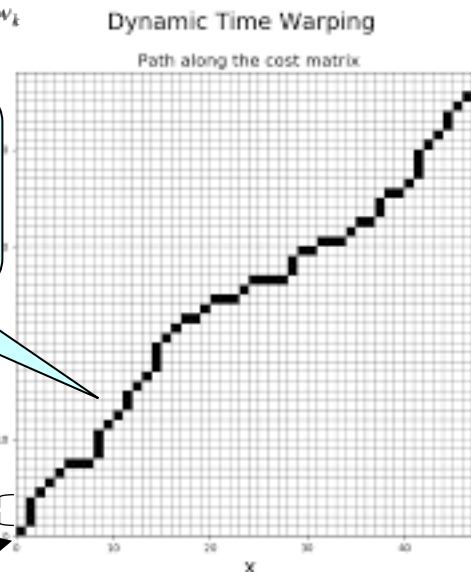
Boundary conditions

Continuity

Monotonicity

Band

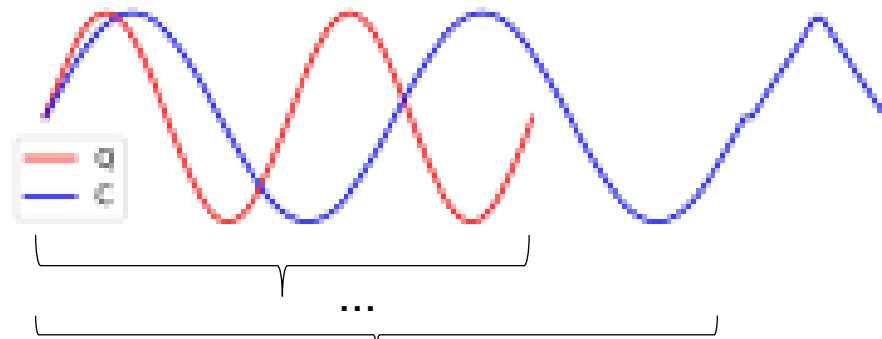
Allowed region



Nearest Neighbor Interpolation and Uniform Scaling

Definition 5 (Nearest Neighbor Interpolation). Given a time series T of length n and an integer L , Nearest Neighbor Interpolation scales T into T^L as follows:

$$T_j^L = T_{\lceil n(j/L) \rceil} \quad \text{where } 1 \leq j \leq L \quad (4)$$



A 4-prefix
of T

$$T = 1, 2, \dots, 6$$

$$\begin{aligned} T(1:4)^8 &= T_{\lceil 4(1/8) \rceil}, T_{\lceil 4(2/8) \rceil}, T_{\lceil 4(3/8) \rceil}, \dots, T_{\lceil 4(8/8) \rceil} \\ &= T_1, T_1, T_2, \dots, T_4 \\ &= 1, 1, 2, \dots, 4. \end{aligned}$$

Interpolate
to 8

Nearest neighbor interpolation
to find the best prefix

$$\text{len} = \lceil m/l \rceil \sim \min(\lfloor lm \rfloor, n)$$

Definition 6 (Uniform Scaling (US) [18]). Given two series Q and C , of length m and n respectively, and a scaling factor bound l , where $l \geq 1$. Let $C(1:k)$ be the prefix of C , where $\lceil m/l \rceil \leq k \leq \min(\lfloor lm \rfloor, n)$, and $C(1:k)^L$ be a rescaled version of $C(1:k)$ with length L , where $L = \min(\lfloor lm \rfloor, n)$. L is called the alignment factor. $\min(\lfloor lm \rfloor, n)$ is the largest alignment factor.

Find the
best prefix

$$\text{US}(Q, C, l, L) = \min_{k=\lceil m/l \rceil}^{\min(\lfloor lm \rfloor, n)} \text{ED}(Q^L, C(1:k)^L) \quad (5)$$

Lower bound

- A number that is fast to compute and lower bound from below to the target function

LB_{Kim}

$$LB_{Kim} = \max \left\{ \begin{array}{l} d(q_1, c_1) \\ d(q_{-1}, c_{-1}) \\ d(q_{\max}, c_{\max}) \\ d(q_{\min}, c_{\min}) \end{array} \right\}$$

∴ boundary condition

∴ A pt in Q must map to a pt in C

LB_{Shen}

$$q_j = (q_{\max(1, \lfloor j/l \rfloor - r)}, \dots, q_{\min(\lfloor j \rfloor + r, m)})$$

$$LB_{Shen}(Q, C) = d(c_1, q_1) + \sum_{j=2}^{n-1} \delta(c_j, q_j) + d(c_n, q_n)$$

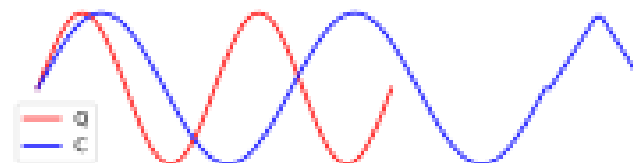
LB_{Keogh}

$$\left. \begin{array}{l} U_j^Q = \max(q_{\max(1, i-r)} : q_{\min(j+r, m)}) \\ L_j^Q = \min(q_{\max(1, i-r)} : q_{\min(j+r, m)}) \end{array} \right\}$$

Form an envelope

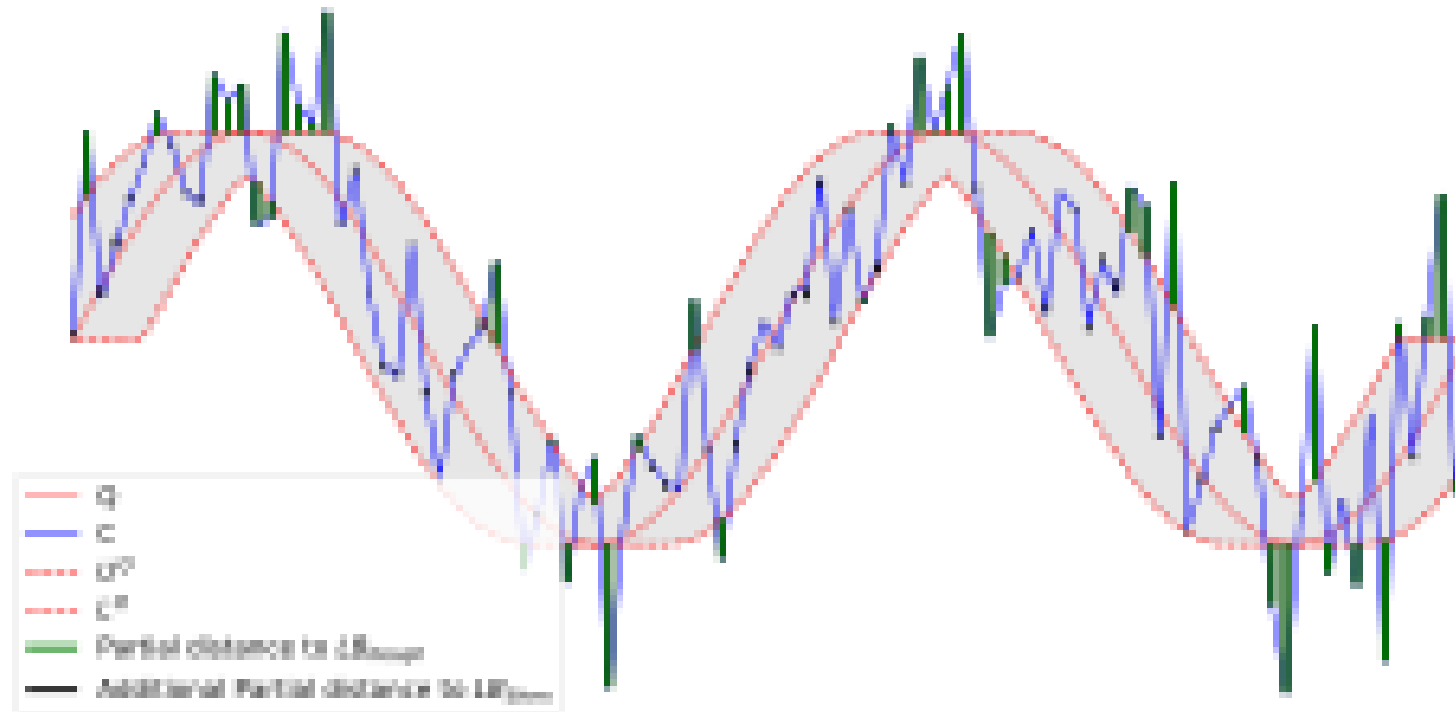
$$LB_{Keogh}(Q, C) = \sum_{j=1}^n \begin{cases} d(c_j, U_j^Q) & \text{if } c_j > U_j^Q \\ d(c_j, L_j^Q) & \text{if } c_j < L_j^Q \\ 0 & \text{otherwise} \end{cases}$$

Theorem 1. For any $\lfloor m/l \rfloor \leq k \leq \min(\lfloor lm \rfloor, n)$, $DTW_r(Q^{\min(\lfloor lm \rfloor, n)}, C(1 : k)^{\min(\lfloor lm \rfloor, n)})$ is always lower bounded by $\sum_{j=1}^k \delta(c_j, q_j)$.



We can grow the LB incrementally

Visualization





Use DTW as the core in US => USDTW

US

$$US(Q, C, l, L) = \min_{k=\lceil m/l \rceil}^{\min(\lfloor lw \rfloor, n)} ED(Q^L, C(1:k)^L)$$

USDTW

$$USDTW_r(Q, C, l, L) =$$

$$\min_{k=\lceil m/l \rceil}^{\min(\lfloor lw \rfloor, n)} DTW_r(Q^L, C(1:k)^L)$$

prefix

DTW parameter

How to deal with more than one scaling factor? Our framework PS

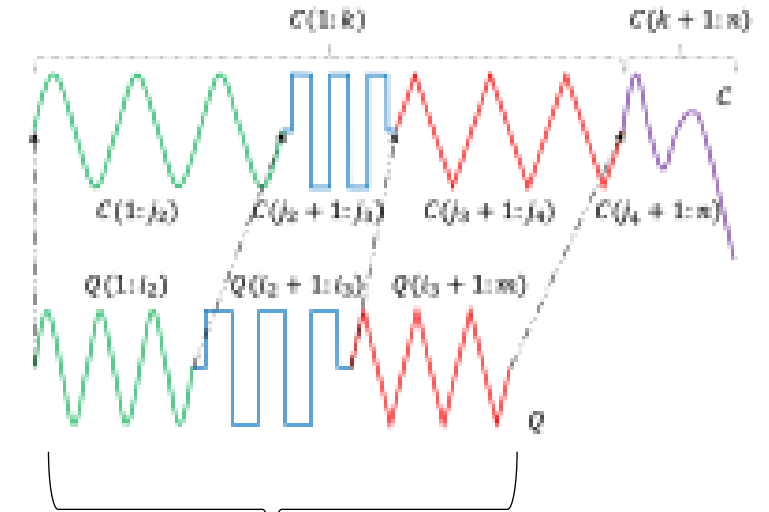
$$\text{USDW}_r(Q, C, l, L) =$$

$$\min_{k=\lfloor \frac{m}{l} \rfloor}^{\min(\lfloor \frac{m}{l} \rfloor, n)} \text{DTW}_r(Q^L, C(1:k)^L)$$

$$\text{PSDTW}_r(Q, C, l, L, P) =$$

$$\min_{\substack{l_1 < l_2 < \dots < l_{p+1} \\ j_1 < j_2 < \dots < j_{p+1}}} \sum_{p=1}^P \text{DTW}_r(Q(l_p + 1 : l_{p+1})^L,$$

$$C(j_p + 1 : j_{p+1})^L)$$



Only focus on the prefix

A summation of distances between different pairs

Its DP

Problem

$$\text{PSDTW}_r(Q, C, l, L, P) =$$

$$\min_{\substack{l_1 < l_2 < \dots < l_{p+1} \\ j_1 < j_2 < \dots < j_{p+1}}} \sum_{p=1}^P \text{DTW}_r(Q(l_p + 1 : l_{p+1})^L, C(j_p + 1 : j_{p+1})^L)$$

$$C(j_p + 1 : j_{p+1})^L$$

The first i point

DP

$$D[i, j, p] = \min_{\substack{l' < i \\ j' < j}} \left\{ D[l', j', p-1] \right.$$

$$\left. + \text{DTW}_r(Q(l' + 1 : i)^L, C(j' + 1 : j)^L) \right\}$$

Similar to $O(n^2)$ for DTW, it needs to fill the table of size $p \times n \times m$

The min operator takes $O(mn)$

$O(n)$ for ED or $O(n^2)$ for DTW

Algorithm 1 Naive PSDTW

Input: Query series Q , Candidate series C , DTW constraint parameter r (in fraction), Number of pieces P , Scaling parameter L

Output: Distance Matrix D of size $(m+1) \times (n+1) \times (P+1)$

```

1: Initialize  $D$  with  $\infty$ 
2:  $D[0, 0, 0] \leftarrow 0$ 
3: for  $p \leftarrow 1$  to  $P$  do
4:   for  $i \leftarrow 1$  to  $m$  do
5:     for  $j \leftarrow 1$  to  $n$  do
6:        $D[i, j, p] \leftarrow \min_{l' < i, j' < j} \{ D[l', j', p-1] + \text{DTW}_r(Q(l' + 1 : i)^L, C(j' + 1 : j)^L) \}$ 
7: return  $D[m, n, P]$ 

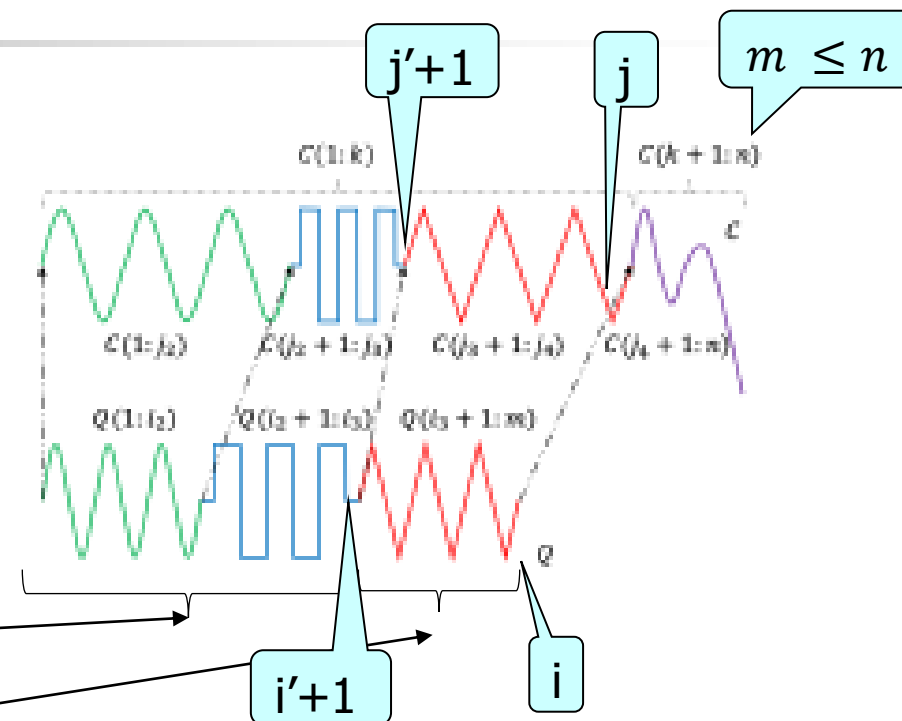
```

Algorithm 2 Line 6 in Algorithm 1

```

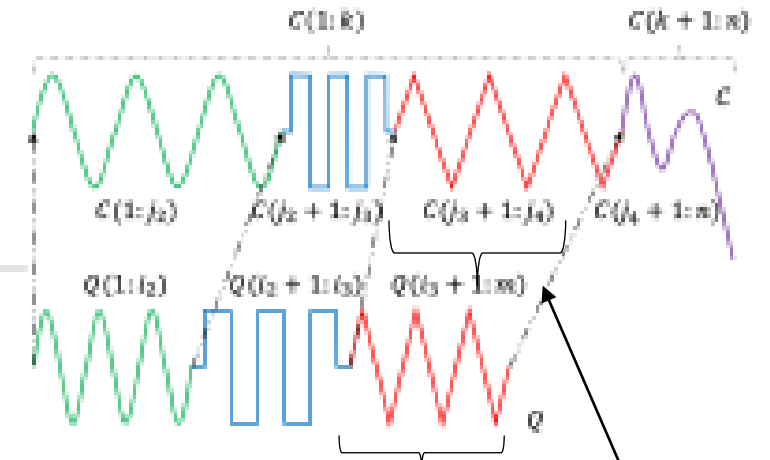
1: for  $l' \leftarrow 0$  to  $i-1$  do
2:   for  $j' \leftarrow 0$  to  $j-1$  do
3:      $\text{dist}_{\text{prev}} \leftarrow D[l', j', p-1]$ 
4:      $\text{dist}_{\text{seg}} \leftarrow \text{DTW}_r(Q(l' + 1 : i)^L, C(j' + 1 : j)^L)$ 
5:      $D[i, j, p] \leftarrow \min(D[i, j, p], \text{dist}_{\text{prev}} + \text{dist}_{\text{seg}})$ 

```



Speed it up

- Similar to bend
- We set up the allowed length



Algorithm 3 Initialization of PSDTW

- 1: $L_{\min}^Q \leftarrow \lceil (m/P)/\sqrt{l} \rceil$, $L_{\max}^Q \leftarrow \lfloor (m/P)\sqrt{l} \rfloor$ \triangleright "g" refers to "global".
- 2: $L_{\min}^C \leftarrow \lceil (n/P)/\sqrt{l} \rceil$, $L_{\max}^C \leftarrow \lfloor (n/P)\sqrt{l} \rfloor$
- 3: $L = \max(L_{\max}^Q, L_{\min}^C)$
- 4: Initialize D of size $(m+1) \times (n+1) \times (P+1)$ with ∞
- 5: $D[0, 0, 0] \leftarrow 0$

Allowed length for a segment in Q

Allowed length for a segment in C



Early Abandoning, Parallel Computing

- In nearest neighbor search, we only need to find the nearest search, suppose the current distance between Q and the NN is bsf.
- For $D[i, j, p]$, once the result of the subquestion is larger than bsf, we don't need to consider it anymore

$$D[i, j, p] = \min_{\substack{i' < i \\ j' < j}} \left\{ D[i', j', p-1] + \text{DTW}_r(Q(i'+1:j^L), C(j'+1:j^L)) \right\}$$

Depends on the state of p-1

So we can run in parallel

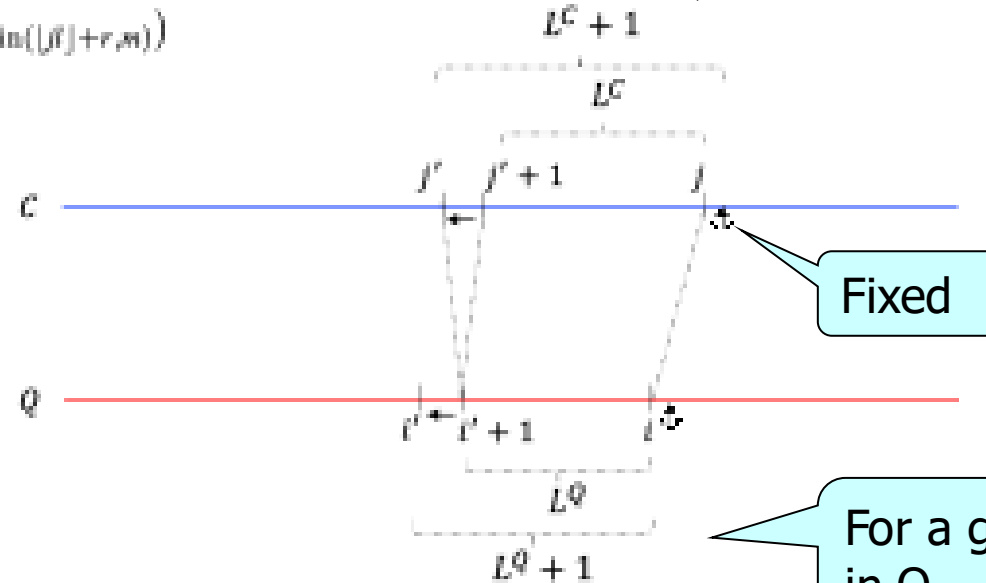
Speed up DTW

It is the same in the case of USDTW

It will compare it with a growing subsequence in C

$$a_j = (q_{\max(1, \lfloor j/l \rfloor - r)}, \dots, q_{\min(\lfloor j \rfloor + r, n)})$$

Possible reach



For a new subsequence of length $L^Q + 1$, we can reuse the previous possible reach to compute a new set of possible reach

For a given L^Q -subsequence in Q



Guided Distance

- Once we computed PSED or PSDTW, we also have the cut (i.e., how the pairs should be aligned). Based on this cut, we can apply other distance measures within the pairs

$$M^{\text{PSED}}(Q, C) = \sum_{p=1}^P M(Q_p^L, C_p^L) \quad (16)$$

Up scale interpolation

, where $Q_p = Q(i_p + 1 : i_{p+1})$ and $C_p = C(j_p + 1 : j_{p+1})$.

Returned by PSD

Querying

- GunPoint dataset
 - Two classes: Gun VS Point

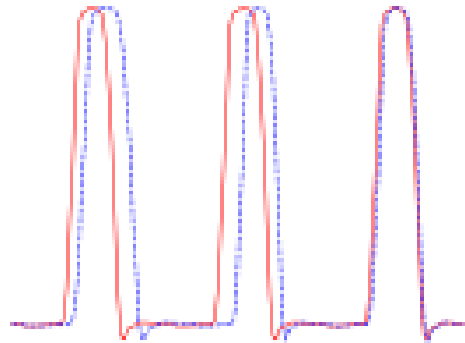
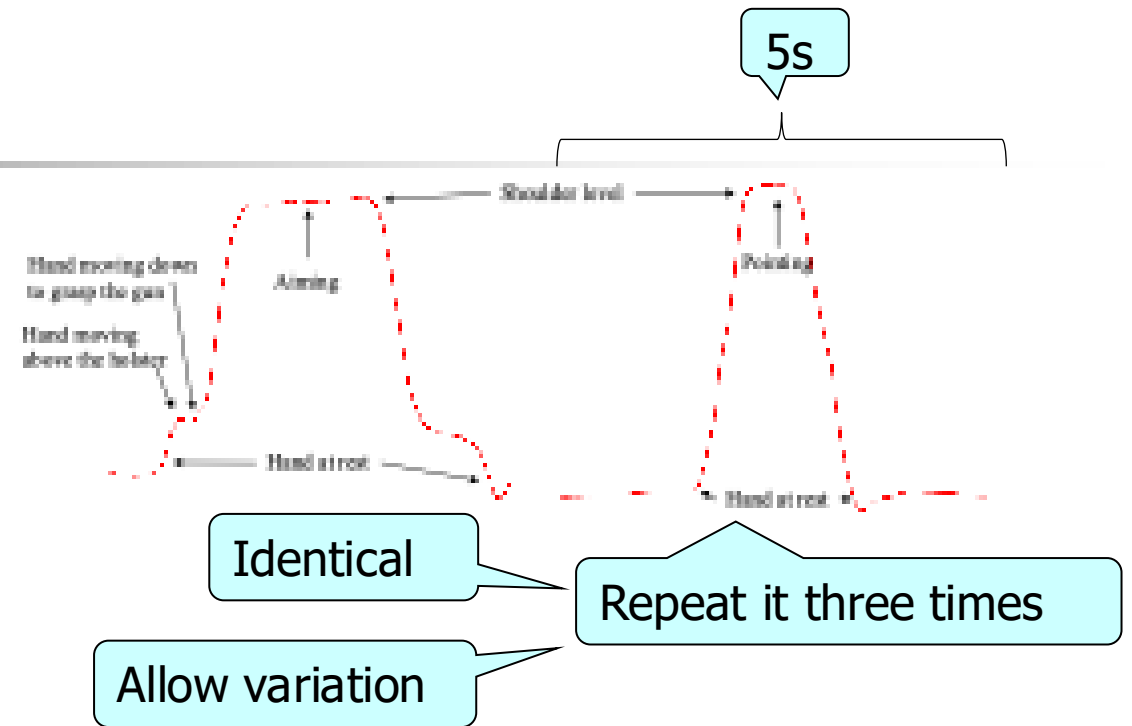


FIGURE 9. The red solid time series shows an instant in the target set. The blue dashed time series shows an instant in the query set.



Given the blue one, can we retrieve the red one

Results

Our method can achieve this kind of invariance

P	l	ED		DTW [13]		ADTW [29]		DDTW [30]		shapeDTW [31]		WDDTW [32]		WDTW [32]		PSED		PSDTW	
		P@1	P@3	P@1	P@3	P@1	P@3	P@1	P@3	P@1	P@3	P@1	P@3	P@1	P@3	P@1	P@3	P@1	P@3
2	1.25	0.30	0.56	0.82	1.00	0.54	0.84	0.84	0.92	0.96	1.00	0.88	0.96	0.82	0.98	0.98	1.00	0.86	1.00
	1.50	0.14	0.36	0.88	1.00	0.38	0.64	0.78	0.94	1.00	1.00	0.80	0.92	0.88	0.96	1.00	1.00	0.96	1.00
	1.75	0.16	0.34	0.82	1.00	0.38	0.66	0.64	0.80	0.90	1.00	0.66	0.80	0.82	0.96	1.00	1.00	0.94	1.00
	2.00	0.12	0.24	0.84	0.98	0.34	0.66	0.72	0.88	0.96	1.00	0.68	0.86	0.82	0.98	1.00	1.00	0.96	1.00
3	1.25	0.30	0.50	0.84	0.92	0.70	0.88	0.80	0.92	0.96	0.98	0.84	0.94	0.86	0.98	0.94	1.00	0.88	0.96
	1.50	0.10	0.28	0.84	0.96	0.60	0.78	0.66	0.86	0.90	1.00	0.72	0.86	0.84	0.98	0.96	1.00	0.86	0.96
	1.75	0.10	0.28	<u>0.88</u>	1.00	0.42	0.78	0.72	0.88	0.76	0.94	0.74	0.88	<u>0.88</u>	1.00	1.00	1.00	0.88	0.98
	2.00	0.02	0.12	<u>0.86</u>	0.94	0.38	0.52	0.60	0.78	0.70	0.94	0.60	0.80	0.82	0.92	0.98	1.00	<u>0.86</u>	0.98
4	1.25	0.34	0.50	0.70	0.86	0.68	0.88	0.60	0.78	0.70	0.90	0.64	0.80	0.70	0.86	0.86	0.94	<u>0.72</u>	0.88
	1.50	0.22	0.38	0.64	0.80	0.60	0.92	0.52	0.64	0.68	0.90	0.52	0.66	0.64	0.82	0.86	0.98	<u>0.72</u>	0.82
	1.75	0.16	0.30	0.72	0.92	0.56	0.80	0.60	0.78	0.56	0.80	0.56	0.80	0.70	0.88	0.92	1.00	<u>0.80</u>	0.94
	2.00	0.06	0.12	0.58	0.78	0.50	0.72	0.46	0.64	0.50	0.82	0.50	0.64	0.56	0.82	0.88	0.98	<u>0.64</u>	0.88

It means worsen the result

P	l	DTW ^{PSED}		ADTW ^{PSED}		DDTW ^{PSED}		shapeDTW ^{PSED}		WDDTW ^{PSED}		WDTW ^{PSED}	
		P@1	P@3	P@1	P@3	P@1	P@3	P@1	P@3	P@1	P@3	P@1	P@3
2	1.25	0.86	1.00	0.98	1.00	0.72	0.92	0.98	1.00	0.74	0.96	0.86	1.00
	1.50	0.88	1.00	1.00	1.00	0.60	0.90	0.98	1.00	0.64	0.90	0.92	1.00
	1.75	0.94	1.00	1.00	1.00	0.84	0.94	1.00	1.00	0.86	0.96	0.94	1.00
	2.00	0.98	1.00	1.00	1.00	0.60	0.78	0.98	1.00	0.66	0.80	0.98	1.00
3	1.25	0.82	0.96	0.94	1.00	0.74	0.90	0.92	0.98	0.76	0.90	0.82	0.96
	1.50	0.86	1.00	0.96	1.00	0.72	0.90	0.94	1.00	0.80	0.90	0.88	1.00
	1.75	0.88	1.00	1.00	1.00	0.68	0.90	1.00	1.00	0.70	0.94	0.90	1.00
	2.00	0.86	0.96	0.98	1.00	0.54	0.86	0.96	1.00	0.62	0.86	0.88	0.96
4	1.25	0.74	0.88	0.86	0.94	0.78	0.80	0.80	0.92	0.78	0.82	0.74	0.88
	1.50	0.72	0.84	0.86	0.98	0.68	0.84	0.84	0.96	0.66	0.84	0.72	0.86
	1.75	0.74	0.96	0.92	1.00	0.54	0.90	0.90	1.00	0.56	0.88	0.74	0.96
	2.00	0.66	0.90	0.88	0.98	0.58	0.82	0.88	0.98	0.62	0.80	0.66	0.90

It helps most of them

All ten datasets

Dataset	ED		DTW		ADTW		DDTW		shapeDTW		WDDTW		WDTW		PSED	
	P@1	P@3	P@1	P@3	P@1	P@3	P@1	P@3	P@1	P@3	P@1	P@3	P@1	P@3	P@1	P@3
SonyAIBORobotSurface1	0.04	0.08	0.51	0.63	<u>0.60</u>	0.74	0.43	0.57	0.20	0.29	0.45	0.58	0.51	0.64	0.67	0.77
ECG200	0.12	0.20	0.70	0.81	<u>0.78</u>	0.83	0.60	0.74	0.59	0.76	0.59	0.73	0.70	0.80	0.81	0.88
MedicalImages	0.08	0.19	0.63	0.78	0.69	0.85	0.52	0.68	0.57	0.75	0.51	0.68	0.62	0.78	0.76	0.88
CBF	0.53	0.67	0.83	1.00	<u>0.90</u>	1.00	0.73	0.87	0.73	0.90	0.77	0.90	<u>0.90</u>	1.00	1.00	1.00
SwedishLeaf	0.07	0.12	0.82	0.92	<u>0.84</u>	0.93	0.70	0.83	0.78	0.91	0.69	0.83	<u>0.83</u>	0.92	0.97	0.99
Plane	0.09	0.14	0.50	0.74	<u>0.59</u>	0.78	0.38	0.64	0.53	0.79	0.37	0.61	0.49	0.75	0.75	0.93
PowerCons	0.26	0.43	0.77	0.98	0.80	1.00	0.77	0.98	0.77	0.99	0.77	0.99	0.77	0.99	0.80	1.00
GunPoint	0.10	0.28	<u>0.84</u>	0.96	0.60	0.78	0.66	0.86	0.90	1.00	0.72	0.86	<u>0.84</u>	0.98	0.96	1.00
Adiac	0.01	0.02	<u>0.37</u>	0.50	0.12	0.21	0.32	0.42	0.27	0.41	0.31	0.41	0.36	0.50	0.57	0.71
Epilepsy	0.18	0.26	0.74	0.88	<u>0.80</u>	0.91	0.65	0.82	0.38	0.47	0.58	0.79	0.70	0.83	0.83	0.91

Dataset	DTW ^{PSED}		ADTW ^{PSED}		DDTW ^{PSED}		shapeDTW ^{PSED}		WDDTW ^{PSED}		WDTW ^{PSED}	
	P@1	P@3	P@1	P@3	P@1	P@3	P@1	P@3	P@1	P@3	P@1	P@3
SonyAIBORobotSurface1	0.53	0.64	0.64	0.76	0.37	0.53	0.65	0.76	0.38	0.53	0.53	0.65
ECG200	0.73	0.84	0.80	0.88	0.54	0.68	0.81	0.88	0.53	0.69	0.73	0.84
MedicalImages	0.66	0.78	0.76	0.88	0.51	0.69	0.74	0.88	0.51	0.70	0.66	0.79
CBF	0.90	1.00	0.93	1.00	0.73	0.93	0.93	0.97	0.70	0.93	0.90	1.00
SwedishLeaf	0.82	0.92	0.97	0.99	0.72	0.87	0.97	0.99	0.73	0.88	0.82	0.92
Plane	0.60	0.80	0.75	0.94	0.50	0.67	0.74	0.90	0.50	0.69	0.59	0.80
GunPoint	0.86	1.00	0.96	1.00	0.72	0.90	0.94	1.00	0.80	0.90	0.88	1.00
PowerCons	0.78	0.99	0.79	1.00	0.79	1.00	0.79	1.00	0.79	1.00	0.78	1.00
Adiac	0.33	0.45	0.57	0.71	0.21	0.34	0.53	0.70	0.20	0.33	0.34	0.45
Epilepsy	0.77	0.88	0.78	0.88	0.66	0.79	0.74	0.89	0.66	0.82	0.80	0.89



Results

- The lower bounding can speed up PSDTW but not PSED
 - Because the overhead of calculating the lower bound outweigh the time saved by pruning
- The following show the effect of using parallel and early abandoning

Name	Size	Length	PSD_vanilla	PSD_parallel_bsf	% distance calls pruned	Speed Up
			Time (s)	Time (s)		
SonyAIBORobotSurfaceI	601	70	697	69	90.40%	10.10X
ECG200	100	96	91	3	91.08%	30.33X
MedicalImages	381	99	2082	40	96.03%	52.05X
CBF	30	128	43	2	75.31%	21.50X
SwedishLeaf	500	128	16601	107	96.24%	155.15X
Plane	105	144	621	6	96.05%	103.50X
PowerCons	180	144	3629	49	82.32%	74.06X
GunPoint	50	150	159	3	89.42%	53.00X
Adiac	390	175	23550	123	96.71%	191.46X
Epilepsy	137	206	13131	336	39.88%	39.08X



Leveraging Nearest Neighbors for Time Series Forecasting with Matrix Profile

- Introduction
- 1. MTSCCleav: a Multivariate Time Series Classification (MTSC)-based Method for Predicting Human Dicer Cleavage
- 2. Scaling with Multiple Scaling Factors in Time Series Searching
- 3. Leveraging Nearest Neighbors for Time Series Forecasting with Matrix Profile
- Conclusion

What is Matrix Profile

Algorithm 1 The brute force approach to compute Matrix Profile

```
1: for  $i \leftarrow 1$  to  $n - m + 1$  do
2:   for  $j \leftarrow 1$  to  $n - m + 1$  do
3:     Compute the z-normalized Euclidean distance between  $t_{i,m}$  and  $t_{j,m}$ .
```

- It tells us the nearest neighbor information of each subsequence in T_A with respect to T_B

- The similarity
- The location

Matrix Profile (MP)

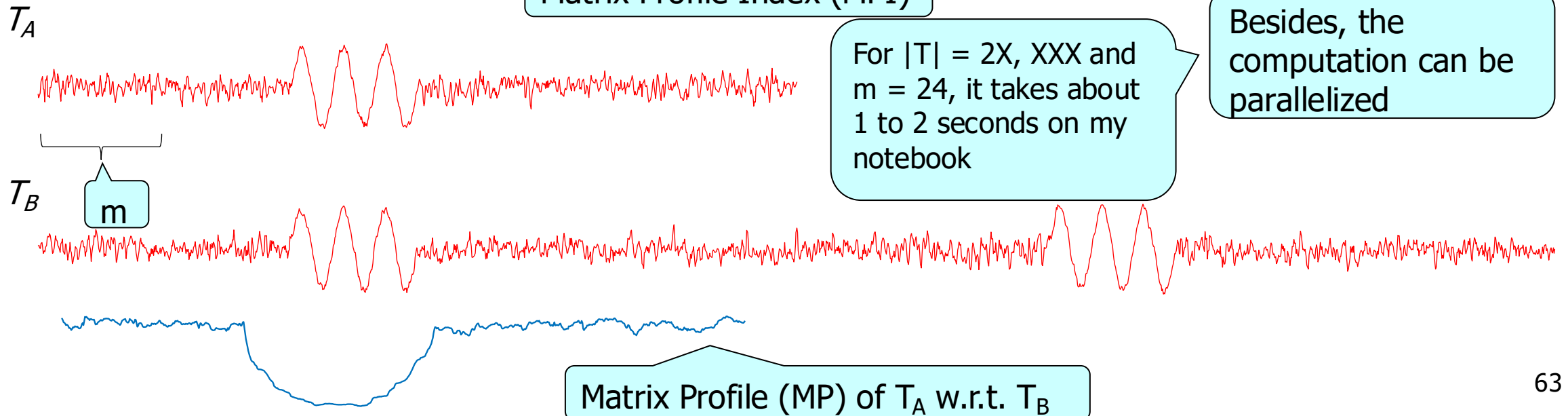
Naïve takes $O(n^3)$

In fact, it can be computed in $O(n^2)$

Matrix Profile Index (MPI)

Besides, the computation can be parallelized

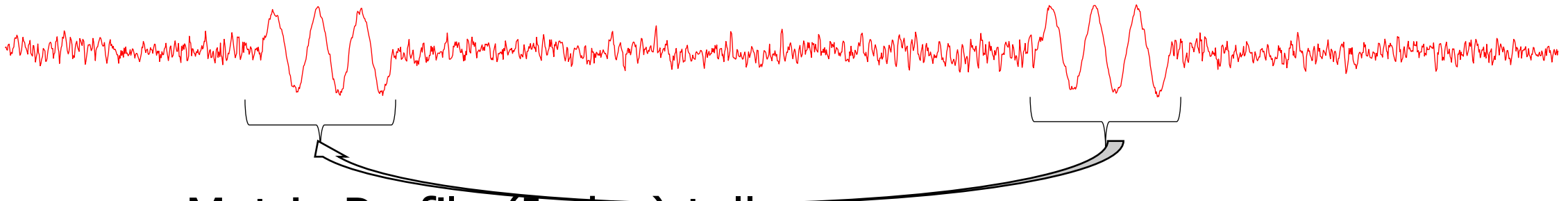
For $|T| = 2X, XXX$ and $m = 24$, it takes about 1 to 2 seconds on my notebook





Finding nearest neighbor within itself

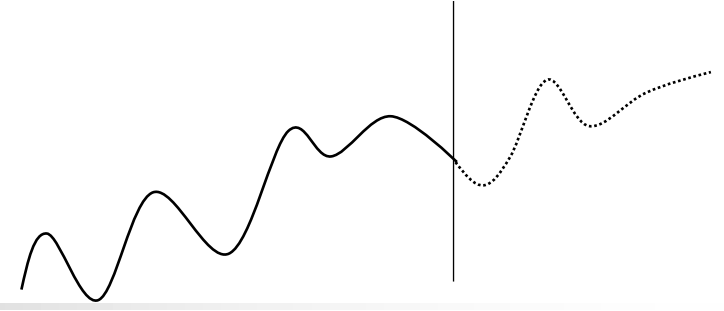
- Suppose now we only have one time series



- Matrix Profile (Index) tells us
 - For each subsequence with a user-given length m
 - What is the similarity in ED between it and its NN
 - Where is its NN

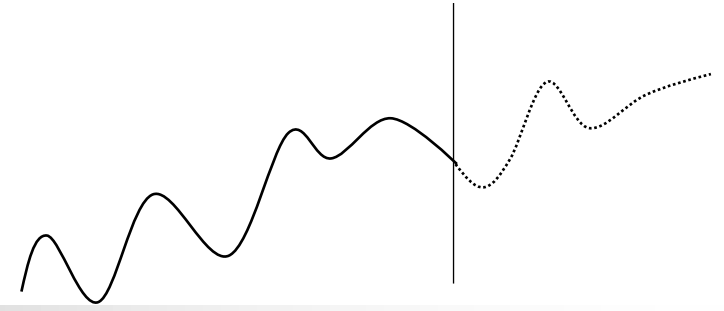


Time Series Forecasting

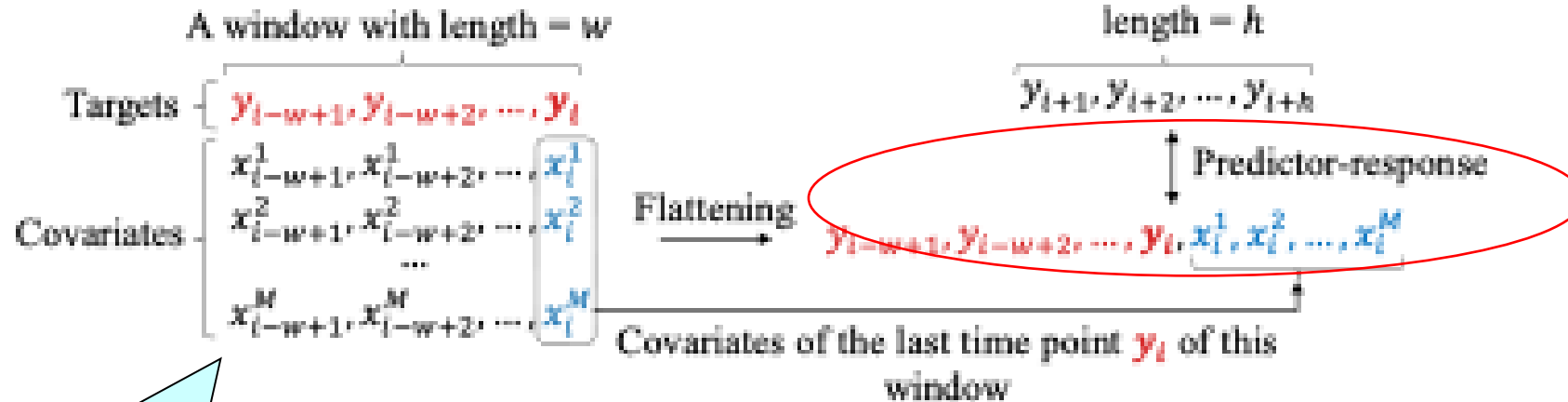


- Given a long time series T , and an integer indicate the split of the training subsequence and test subsequence
 - $|T| = |T_{\text{train}}| + |T_{\text{test}}|$
- We would like to train on T_{train} to predict on T_{test}
- For example, $n = |T| = 26,136$
 - $|T_{\text{train}}| = 25,968$ and $|T_{\text{test}}| = 168$
- But we don't predict the 168 time points in a single shot.
- We are only predicting h at one time
 - Then, we will be given correct answer
 - Teacher forcing

Gradient Boosting Tree



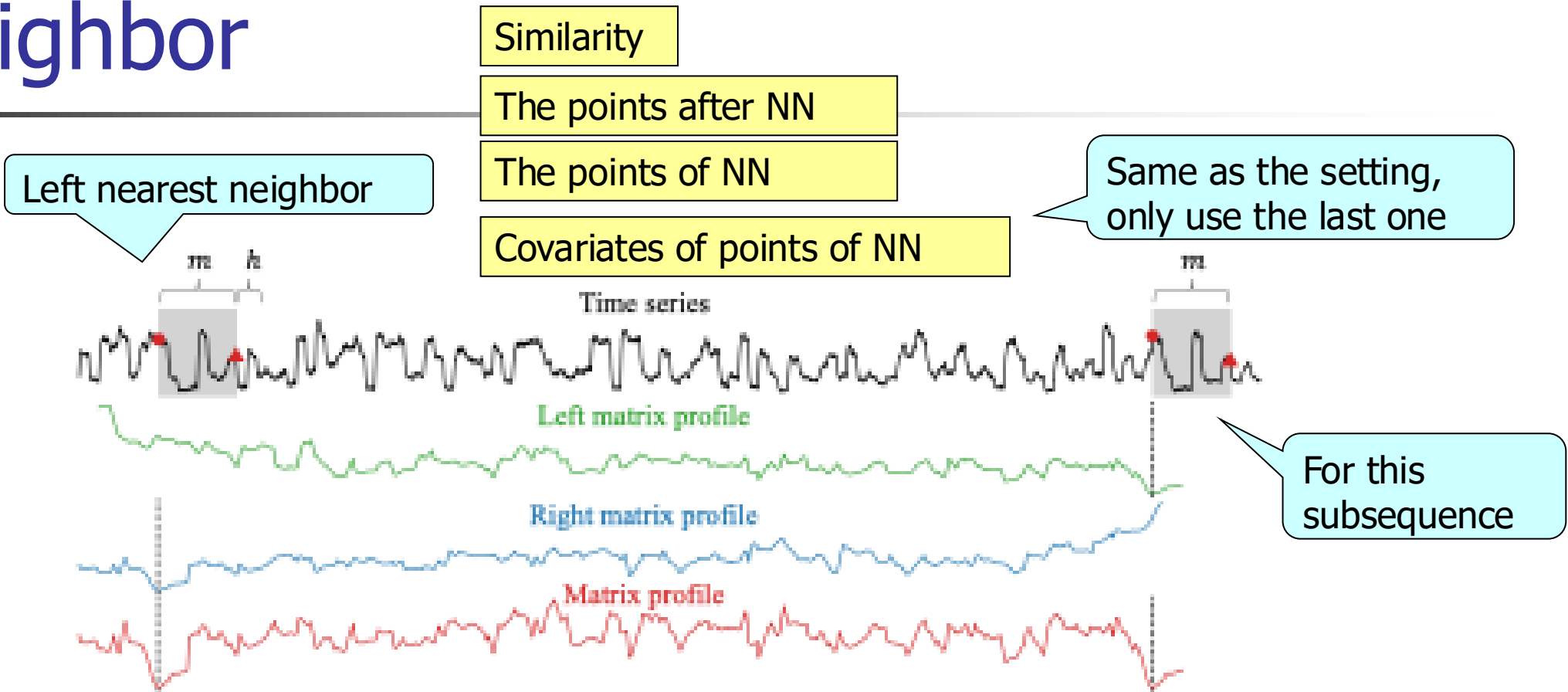
We only consider the look-back window of length = w



day of the week or month,
holiday or not

Only keep the covariates of the
last point of the window

Create more covariates from left nearest neighbor



Datasets

Without covariates

Dataset	Data			Forecasting Task		
	N	n	rate	w, h	$ T_{train} $	$ T_{test} $
Electricity [32]	70	26,136	hourly	24	25,968	168
Traffic [32]	90	10,560	hourly	24	10,392	168
PeMSD7 [32]	228	12,672	15 mins	9	11,232	1,440
Exchange-Rate [1]	8	7,536	daily	24	6,048	1,488

Dataset	Metric	LSTNet* [1]	TRMF* [33]	DARNN* [34]	ARIMA* [35]	GBRT [19]	GBRT-NN	GBRT-NN-S
Electricity [32]	RMSE	1095.309	<u>136.400</u>	404.056	181.210	136.254	142.035	138.354
	WAPE	0.997	0.095	0.343	0.310	0.103	0.101	<u>0.100</u>
	MAE	474.845	53.250	194.449	154.390	57.929	56.464	<u>55.972</u>
Traffic [32]	RMSE	0.042	0.023	0.015	0.044	<u>0.012</u>	0.016	0.008
	WAPE	0.102	0.161	0.132	0.594	0.108	0.079	0.037
	MAE	0.014	0.009	0.007	0.032	0.006	<u>0.004</u>	0.002
PeMSD7 [32]	RMSE	55.405	5.462	5.983	15.357	5.610	5.575	<u>5.557</u>
	WAPE	0.981	0.057	0.060	0.183	0.051	0.051	0.051
	MAE	53.336	3.329	3.526	10.304	2.984	<u>2.965</u>	2.957
Exchange-Rate [1]	RMSE	0.018	0.018	0.025	0.123	0.020	<u>0.019</u>	<u>0.019</u>
	WAPE	0.017	0.015	0.022	0.170	0.016	0.015	0.015
	MAE	0.013	0.011	0.016	0.101	<u>0.012</u>	<u>0.011</u>	<u>0.012</u>

With covariates

Dataset	Metric	DeepGLE* [32]	GBRT [19]	GBRT-NN	GBRT-NN-S
Electricity [32]	RMSE	141.285	132.669	<u>123.591</u>	124.215
	WAPE	0.094	0.0929	0.089	0.089
	MAE	53.006	52.0252	<u>49.606</u>	49.739
Traffic [32]	RMSE	0.028	0.014	0.018	0.009
	WAPE	0.239	0.109	0.101	0.062
	MAE	0.013	0.006	<u>0.006</u>	0.003
PeMSD7 [32]	RMSE	6.490	5.191	5.163	5.156
	WAPE	<u>0.070</u>	0.048	0.048	0.048
	MAE	3.530	2.796	2.772	2.778
Exchange-Rate [1]	RMSE	0.038	0.020	0.019	0.019
	WAPE	0.038	<u>0.016</u>	0.015	0.015
	MAE	0.022	<u>0.012</u>	0.012	0.012

We use h points after NN

It is better to use the raw value than the trend or normalized value

It first decreases with more points used



Q & A

- Thank you!



References

Other references/credits can be found in “Notes” of each slide

- [keogh2022finding]

- E. Keogh, “Finding Approximately Repeated Patterns in Time Series,” 2022. [Online]. Available: <https://u-paris.fr/diip/eamonn-keogh-finding-approximately-repeated-patterns-in-time-series-the-most-useful-and-yet-most-underutilized-primitive-in-time-series-analytics/>

- [keogh2025time]

- E. Keogh, “Time Series Data Mining: A Unifying View (2025),” 2025.

- [mueen2017time]

- A. Mueen and E. Keogh, “Time Series Data Mining Using the Matrix Profile,” KDD, 2017. [Online]. Available: <https://www.cs.ucr.edu/%7Eeamonn/MatrixProfile.html>



Backups
