SIAM Data Mining Conference (SDM), 2013

Time Series Classification under More Realistic Assumptions

Bing Hu

Outline

- Motivation
- Proposed Framework
  - Concepts
  - Algorithms
- Experimental Evaluation
- Conclusion & Future Work

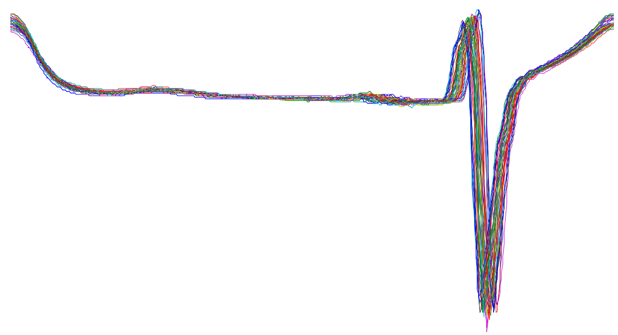
Much of the progress in time series classification from streams is almost *Certainly Optimistic*

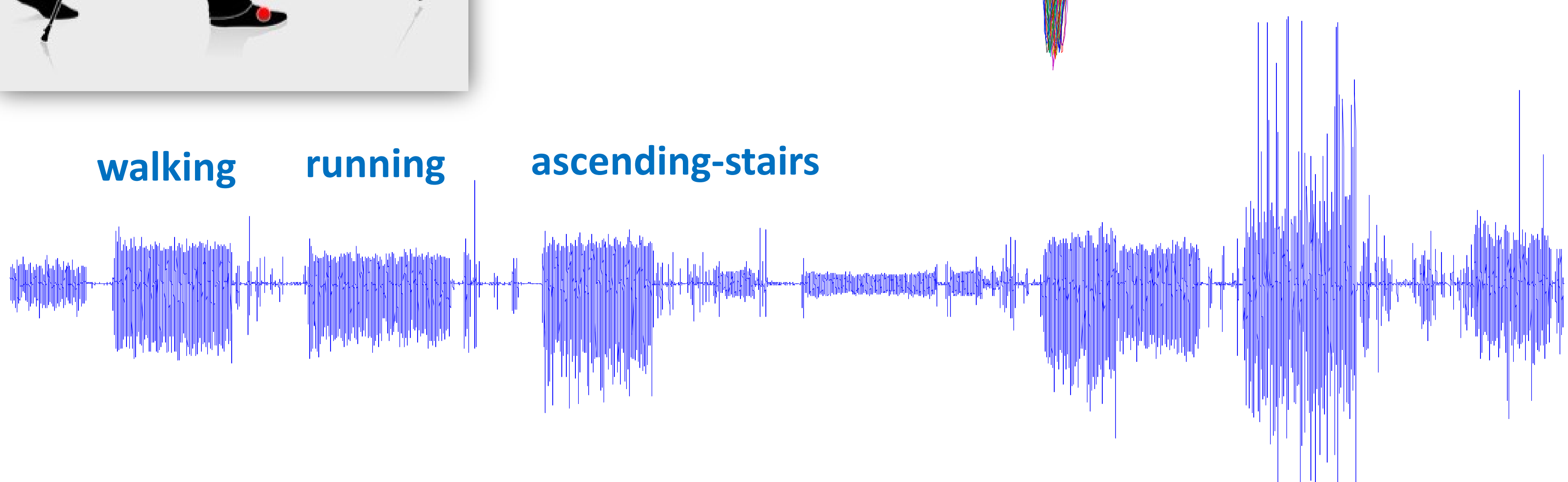
Because they have implicitly or explicitly made *Unrealistic Assumptions*

Assumption (1)

perfectly aligned atomic patterns can be obtained

Individual and complete gait cycles for biometric classification

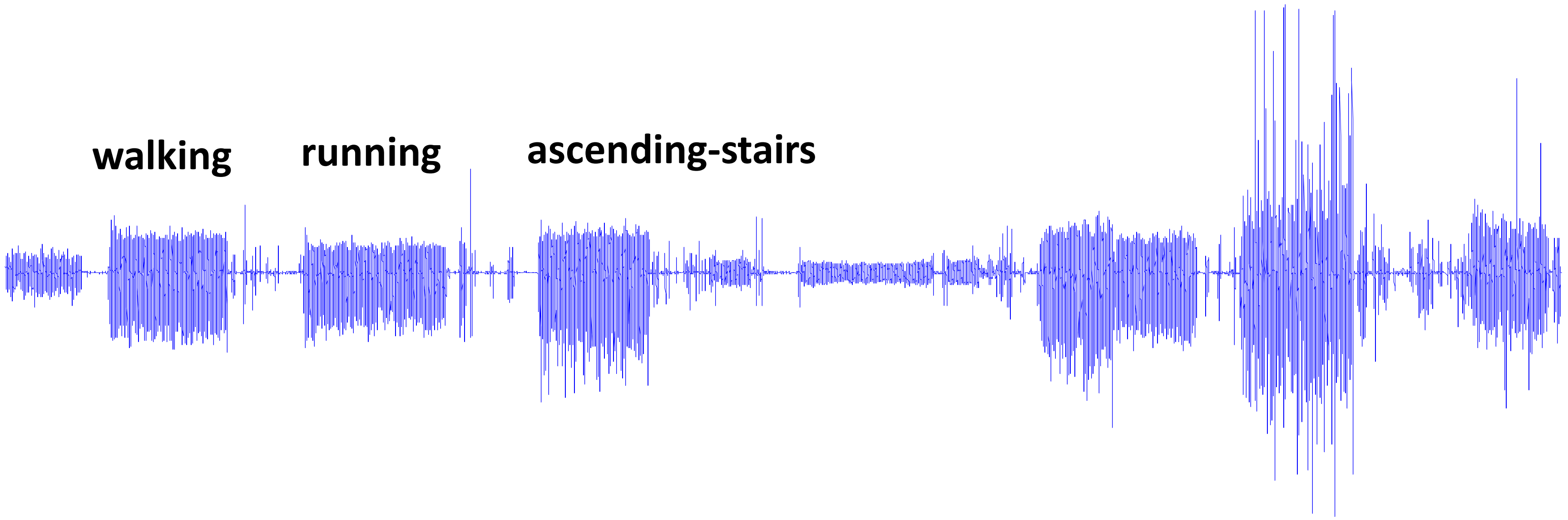




Assumption (1)

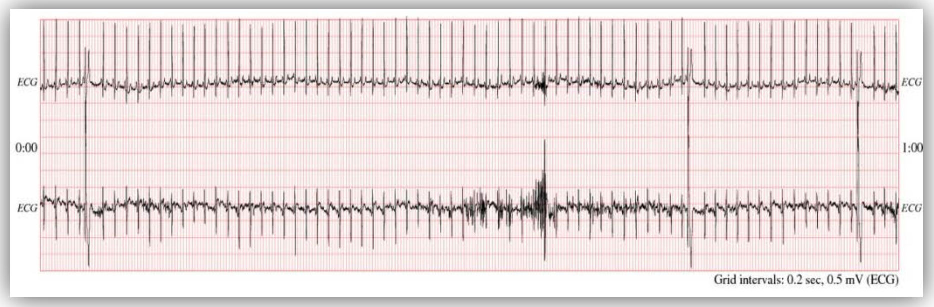
perfectly aligned atomic patterns can be obtained

However, the task of extracting individual gait cycles is not trivial !



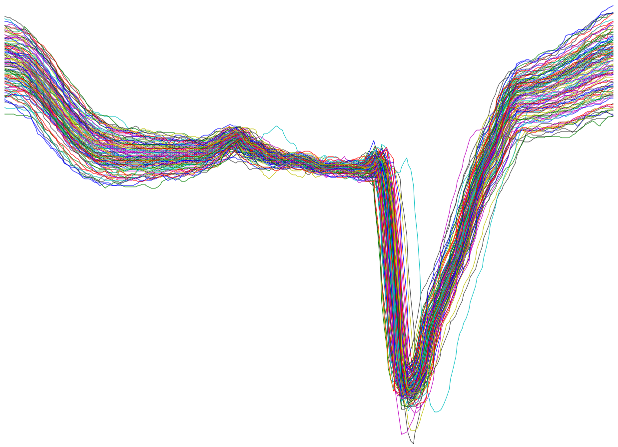
Assumption (2)

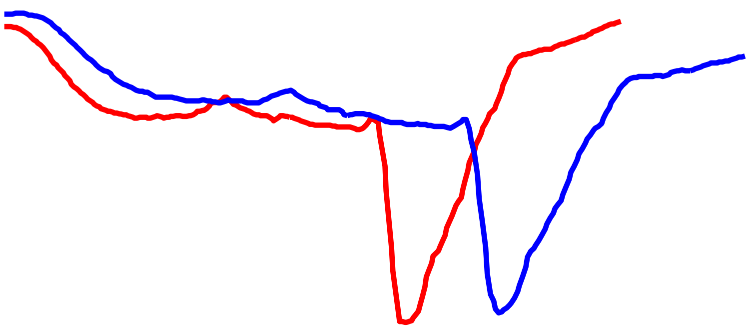
The patterns are all equal length



However,

Heart beat can have different lengths

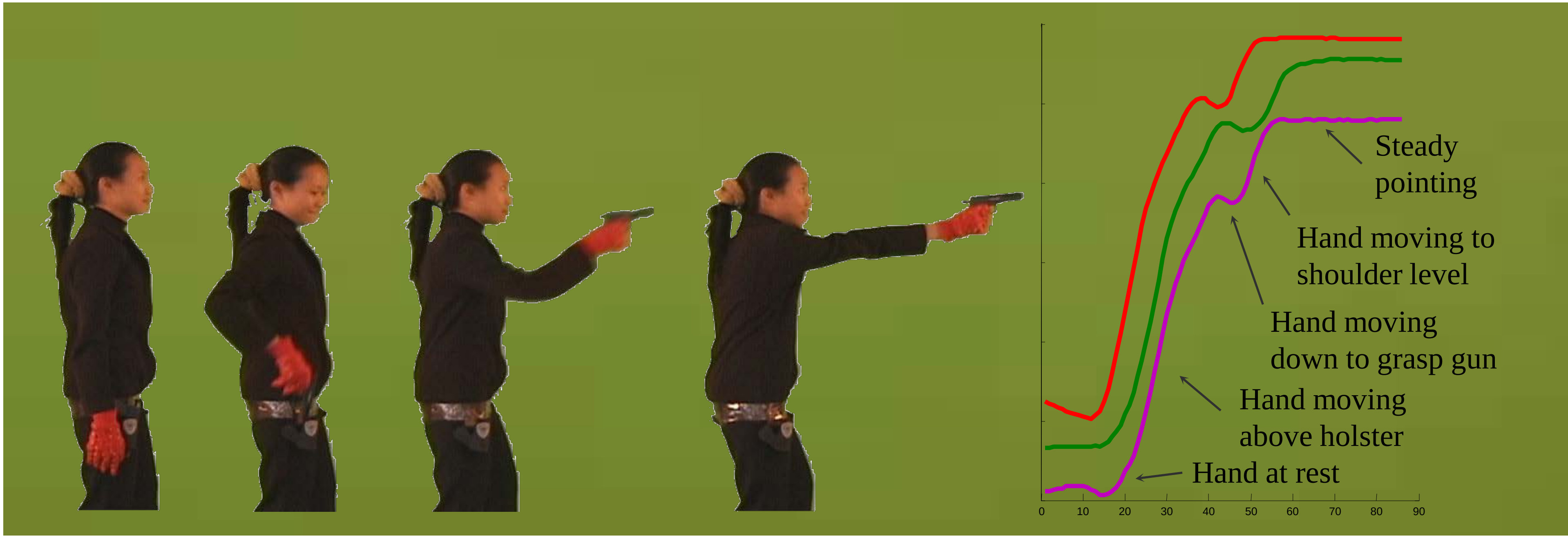




two heart beat of different lengths

Assumption (2)

The patterns are all equal length



Gun/Point problem is probably the ***most studied*** time series classification problem, having appeared in at least one hundred works .

UNREALISTIC !

Assumption (2)

The patterns are all equal length

Contriving of time series datasets seems to be the norm…..

Welcome to the UCR Time Series Classification/Clustering Page

This data resource was funded by an NSF Career Award 0237918, from 2003 to 2008, and continues to be funded through NSF awards 0803410 and 0808770. Partial funding was also made available by a gift from ISCA technologies

This webpage has been created as a public service to the data mining/machine learning community, to encourage *reproducible* research for time series classification and clustering.

Note that the data here is useful for testing *classification / clustering*, and the *accuracy* of indexing techniques. However the datasets are too small to make claims about the *efficiency* of indexing. For this, email Dr. Keogh requesting a free CD-rom of larger datase
you want datasets to test anomaly detection algorithms, many such datasets are here. A comparison of the results below with classic machine learning algorithms is here, thanks to Tony Bagnall and to Weka for this.

Name	First paper or data creator	Number of classes	Size of training set	Size of testing set	Time series Length	1-NN Euclidean Distance	1-NN Best Warping Window DTW (*r*) Note that *r* is the percentage of time series length	1-NN DTW, no Warping Window
Synthetic Control	Pham	6	300 train	300 test	60	0.12	0.017 (6)	0.007
Gun-Point	Ratanamahatana	2	50	150	150	0.087	0.087 (0)	0.093
CBF		3	30	900	128	0.148	0.004 (11)	0.003
Face (all)	Xi	14	560	1,690	131	0.286	0.192 (3)	0.192
OSU Leaf	Gandhi	6	200	242	427	0.483	0.384 (7)	0.409
Swedish Leaf	Soderkvist	15	500	625	128	0.213	0.157 (2)	0.210
50Words	Rath	50	450	455	270	0.369	0.242 (6)	0.310
Trace	Roverso	4	100	100	275	0.24	0.01 (3)	0.0
Two Patterns	Geurts	4	1,000	4,000	128	0.09	0.0015 (4)	0.0
Wafer	Olszewski	2	1,000	6,174	152	0.005	0.005 (1)	0.020
Face (four)	Ratanamahatana	4	24	88	350	0.216	0.114 (2)	0.170
Lightning-2	Eads	2	60	61	637	0.246	0.131 (6)	0.131
Lightning-7	Eads	7	70	73	319	0.425	0.288 (5)	0.274
ECG	Olszewski	2	100	100	96	0.12	0.12 (0)	0.23
Adiac	Jalba	37	390	391	176	0.389	0.391 (3)	0.396
Yoga	Xi	2	300	3000	426	0.170	0.155 (2)	0.164
Fish (readme)	Lee	7	175	175	463	0.217	0.160(4)	0.167
Plane	readme	7	105	105	144	0.038	0.0(5)	0
Car	readme	4	60	60	577	0.267	0.233(1)	0.267
Beef	Tony Bagnall	5	30	30	470	0.467	0.467(0)	0.5
Coffee	Tony Bagnall	2	28	28	286	0.25	0.179(3)	0.179
OliveOil	Tony Bagnall	4	30	30	570	0.133	0.167(1)	0.133

All forty-five time series datasets contain *only equal-length* data

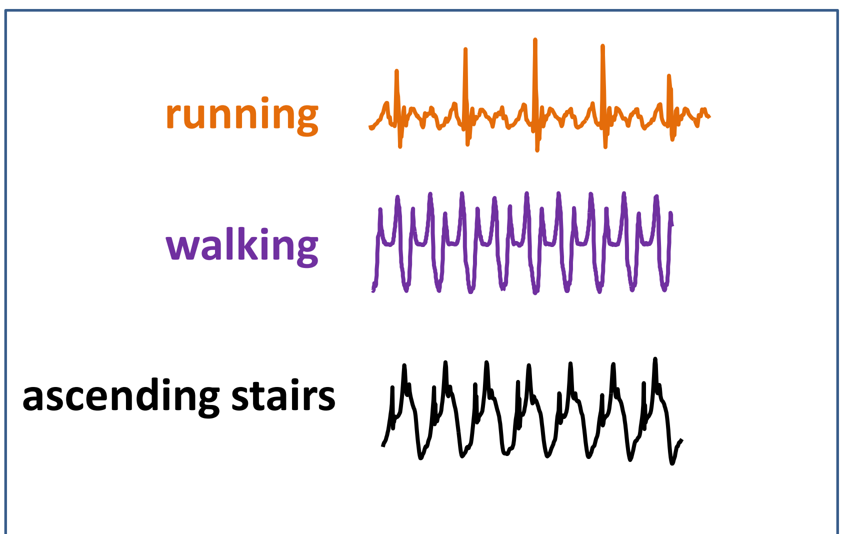
Assumption (3)

Every item that to be classified belongs to *exactly* one of the well-defined classes

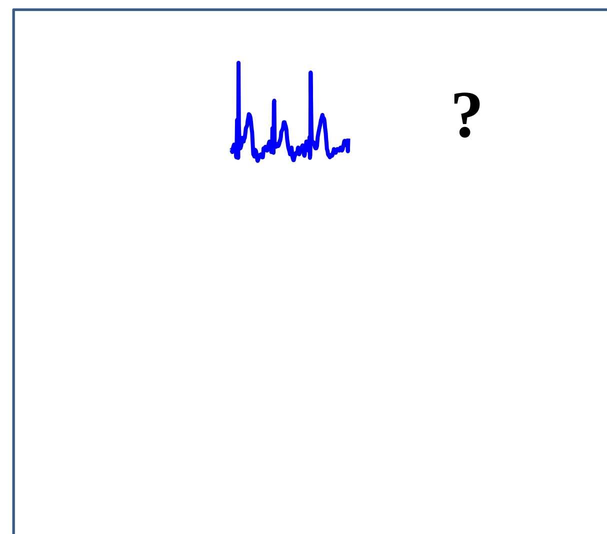
Assumption (3)

Every item that to be classified belongs to *exactly* one of the well-defined classes

training data



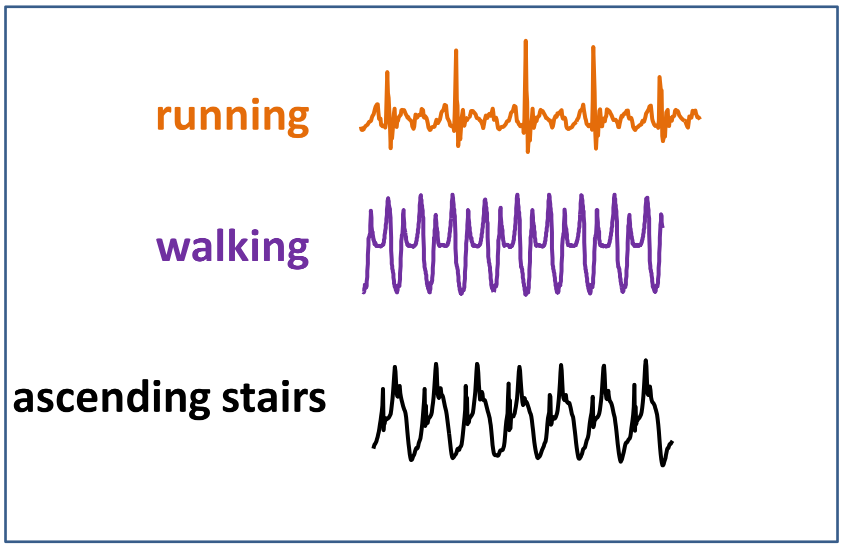
queries



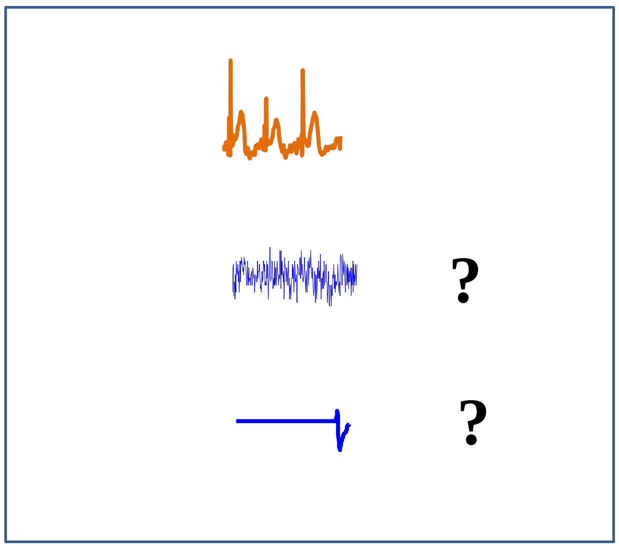
Assumption (3)

Every item that to be classified belongs to *exactly* one of the well-defined classes

training data



queries



A person can not perform `walking` or `running` all the time…
The classification framework must be willing to say **I DO NOT KNOW**

Summary

Most of the literature implicitly or explicitly *assumes* one or more of the following :

Unrealistic Assumptions

- Copious amounts of *perfectly aligned atomic patterns* can be obtained
- The patterns are *all equal length*
- Every item that we attempt to classify *belongs to exactly one of the well-defined classes*

Outline

- Motivation
- Proposed Framework
  - Concepts
  - Algorithms
- Experimental Evaluation
- Conclusion & Future Work

We demonstrate a time series classification framework that does not make ***any*** of these assumptions.

Our Proposal

- **Leverages weakly-labeled data**

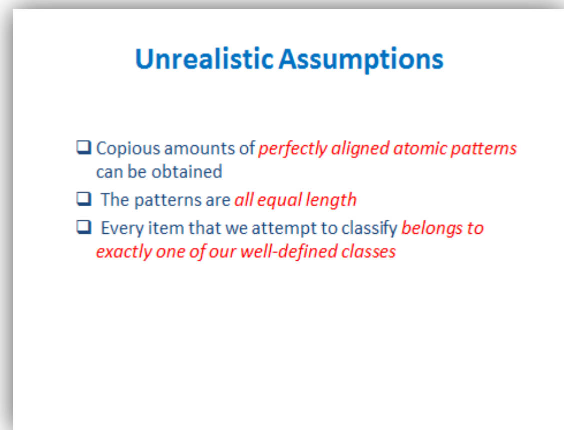
  removes assumption (1) (2)

- **Utilizes a data dictionary**

  removes assumption (1) (2)

- **Exploits rejection threshold**

  removes assumption (3)



Assumptions :

(1) *perfectly* ***aligned atomic patterns***

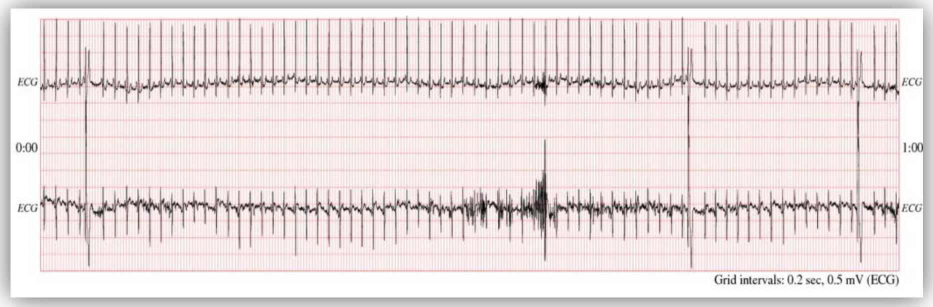
(2) *patterns are* ***all of equal lengths***

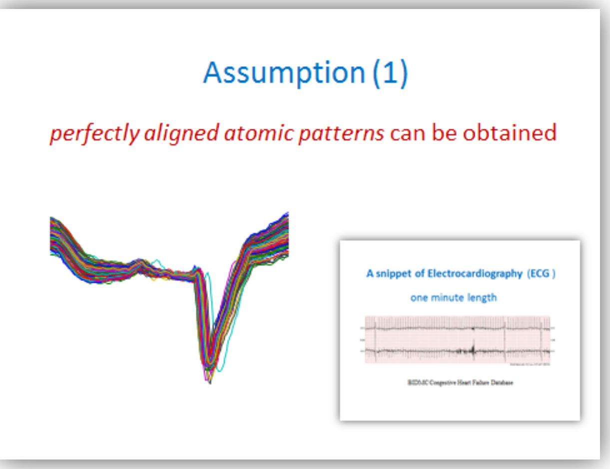
(3) *every item to classify belongs to* ***exactly*** *one of the well-defined classes*

Weakly-Labeled data

such as “*This ten-minute trace of ECG data consists mostly of arrhythmias, and that three-minute trace seems mostly free of them*”

removing assumption (1)

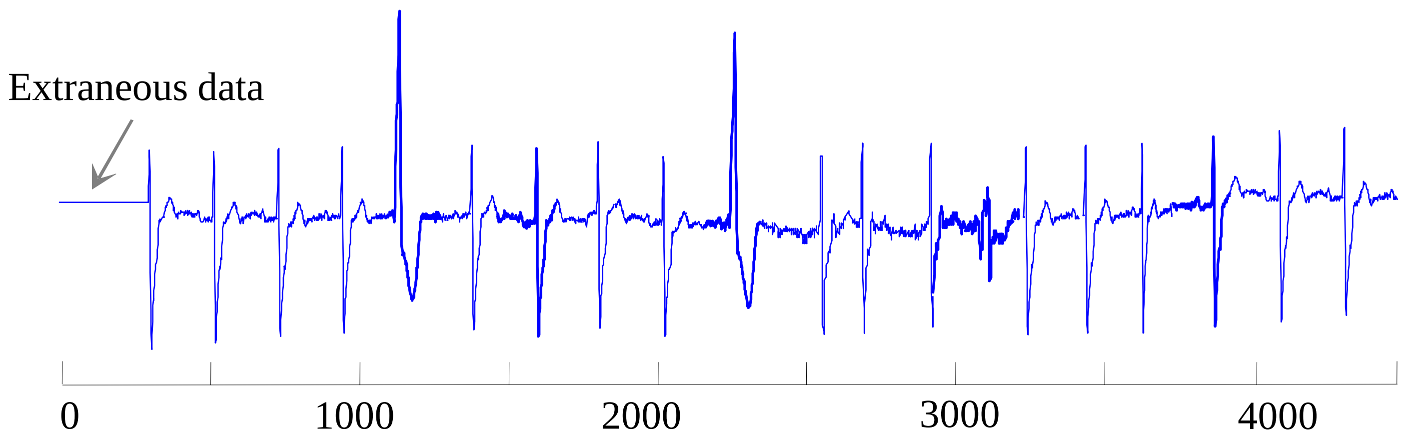




Weakly-Labeled data

- Extraneous/irrelevant sections
- Redundancies

weakly-labeled data from Bob



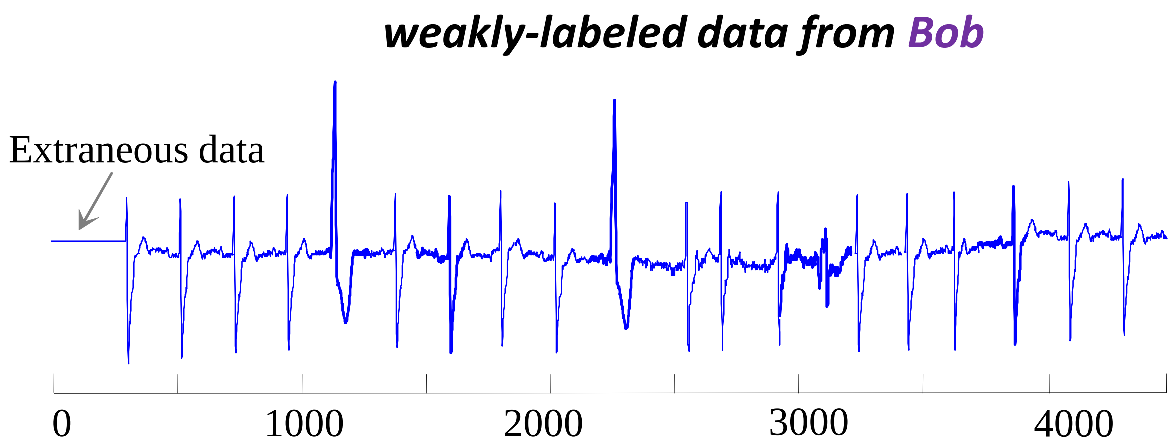
Weakly-Labeled data

How to mitigate the problem of weakly-labeled data?

- Extraneous/irrelevant sections
- Redundancies

Data Dictionary

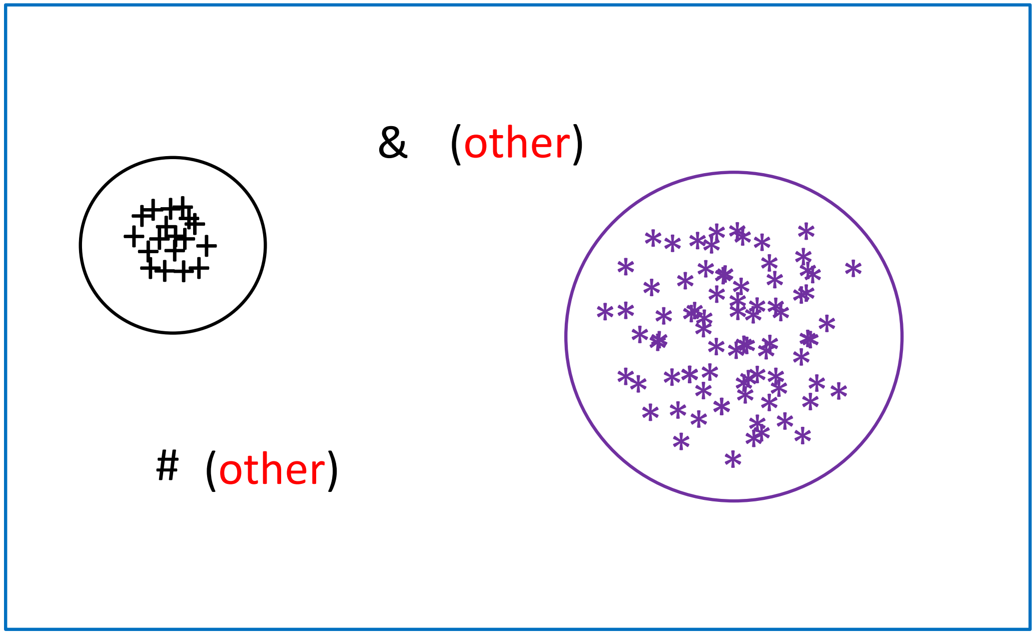
- A (potentially very small) "smart" subset of the training data.
- It spans the concept space.



We want to perform ECG classification between *Bob* and other person's heartbeat

Concept space

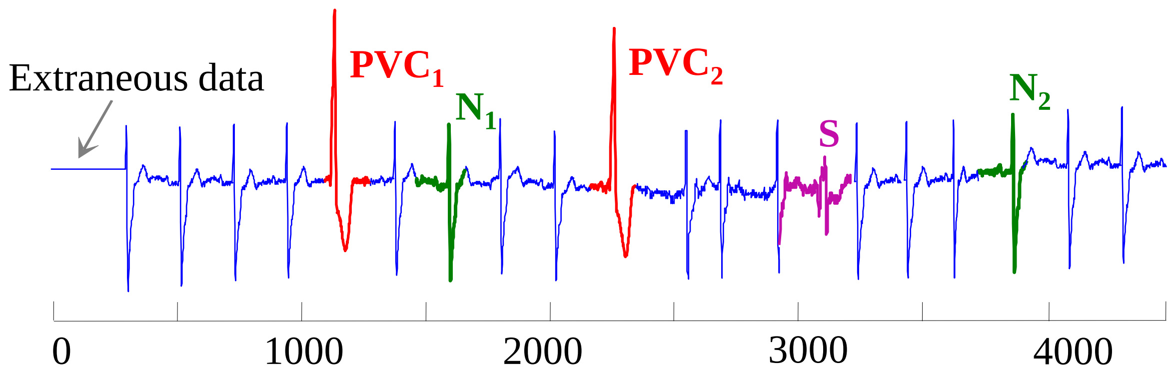
Anything beyond the threshold, it is in other class



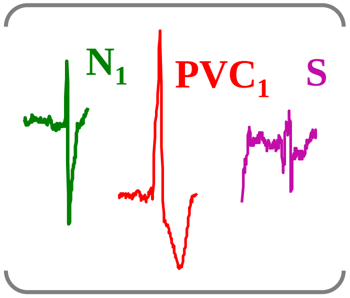
In the above figure, the concept space is one “ * ” and one “+”

Data Dictionary

weakly-labeled data



data dictionary



- ❑ Our algorithm does not know the patterns in advance.
- ❑ We learn those patterns.

PVC: **P**remature **V**entricular **C**ontraction
S: **S**upraventricular Ectopic Atrial
N: **N**ormal ECG

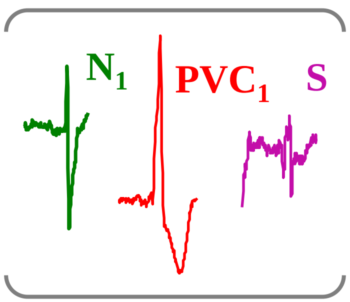
Unrealistic Assumptions

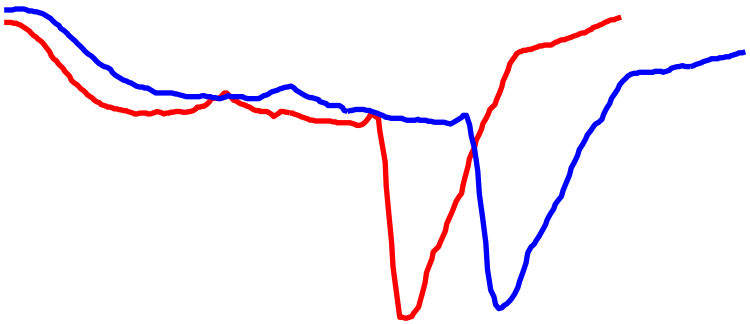
- Copious amounts of *perfectly aligned atomic patterns* can be obtained
- The patterns are *all equal length*
- Every item that we attempt to classify *belongs to exactly one of our well-defined classes*

Data Dictionary

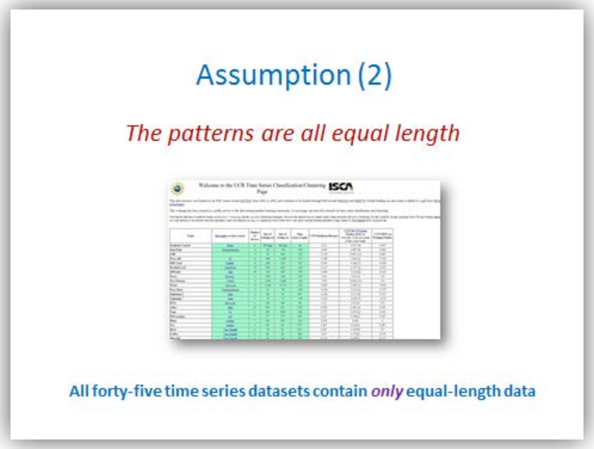
The patterns to be classified can be of different lengths

data dictionary





- leisurely-amble
- normal-paced-walk
- brisk-walk



Unrealistic Assumptions

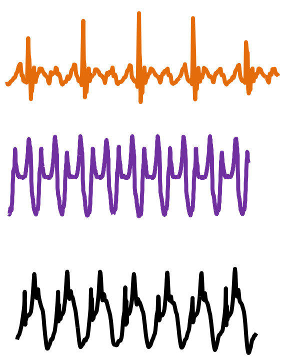
- Copious amounts of *perfectly aligned atomic patterns* can be obtained
- The patterns are *all equal length*
- Every item that we attempt to classify *belongs to exactly one of our well-defined classes*

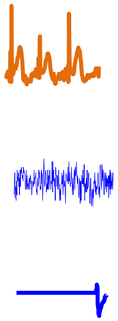
Rejection Threshold

A byproduct of the data dictionary

```
if  NN_Dist of query > threshold
    query is in the other class
```

data dictionary		threshold	queries		
running		7.6		NN_dist < 7.6	running
walking		6.4		NN_dist > 6.4	***other***
ascending stairs		7.3		NN_dist > 7.3	***other***





A person cannot perform `running`, `walking`, `ascending-stairs` all the time. There must exist ***other*** classes.

Desirable Properties of Data Dictionaries

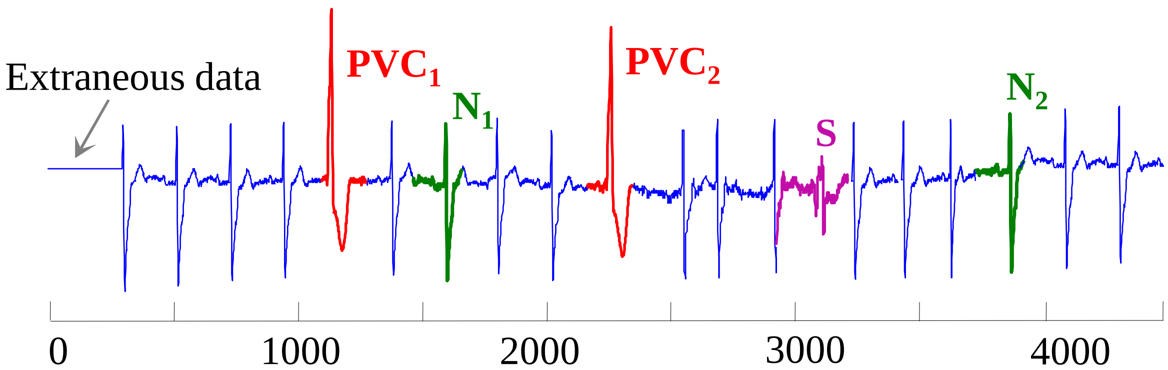
- the classification error rate using D should be ***no worse*** than (***can be better***) using all the training data

Why ?

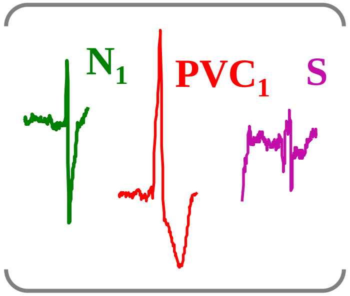
Desirable Properties of Data Dictionaries

This is because the data dictionaries contains less spurious/misleading data.

weakly-labeled data



data dictionary

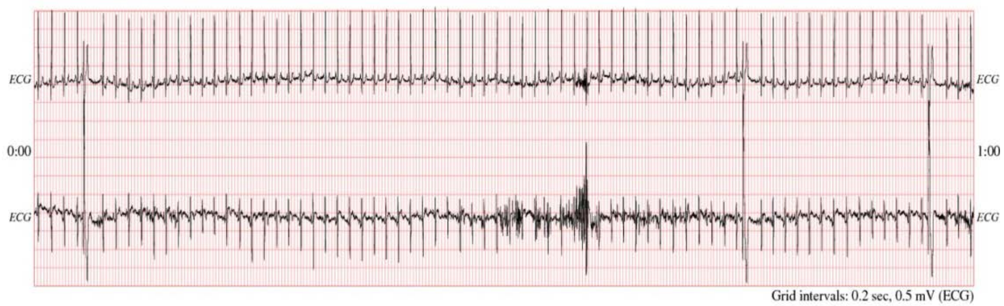


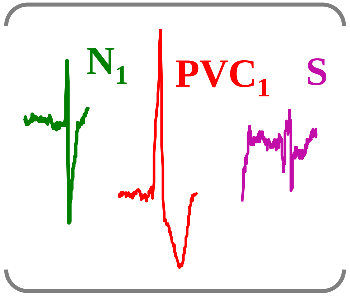
Desirable Properties of Data Dictionaries

D can be a very small percentage of the training data

- ✓ faster running time
- ✓ resource limited device

data dictionary





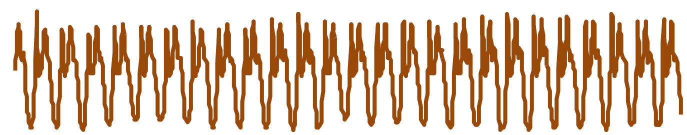
for one hour of ECG data → Data dictionary

Space : **3600Kbits** **20 Kbits**

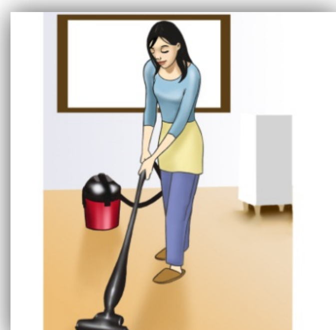
Desirable Properties of Data Dictionaries

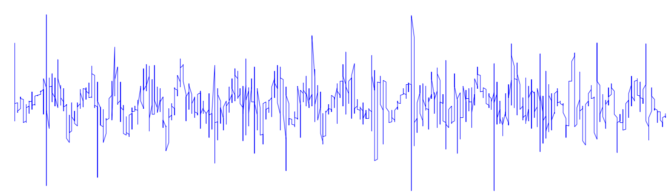
the number of subsequences within each class in **D** can be different

walking



vacuum cleaning



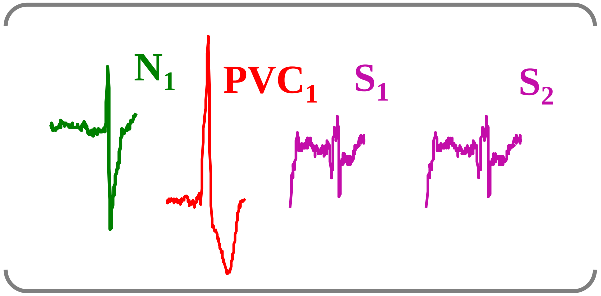


Desirable Properties of Data Dictionaries

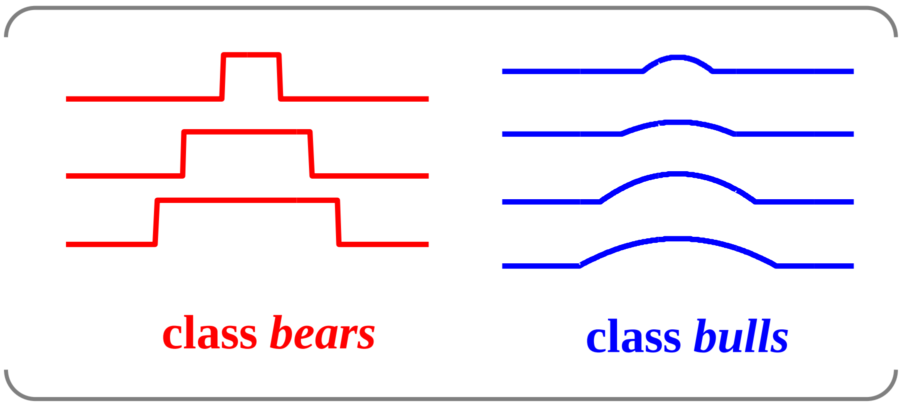
the number of subsequences within each class in **D** can be different

- ✓ For example, if the number of S in **D** is larger than PVC , we can conclude that the variance of S is larger than PVC

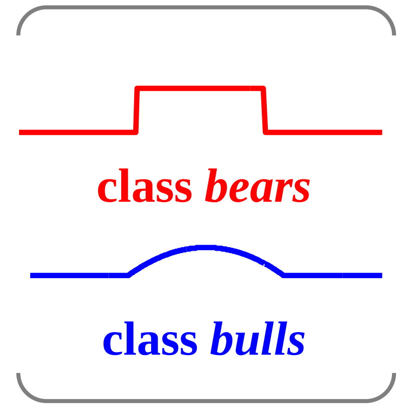
data dictionary



An Additional Insight on Data Redundancy

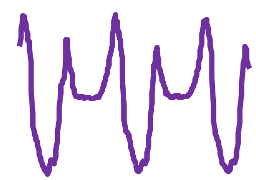


Data dictionary A



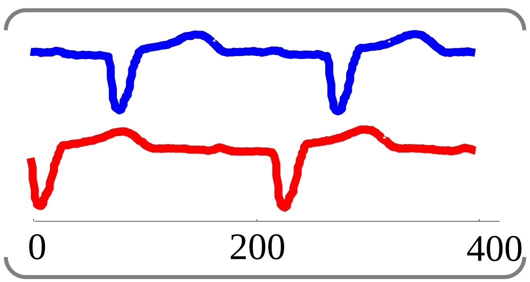
Data dictionary B

- **leisurely-amble**
- **normal-paced-walk**
- **brisk-walk**

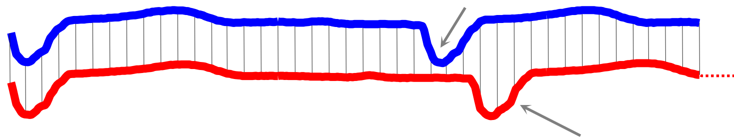


Our Solution : Uniform Scaling

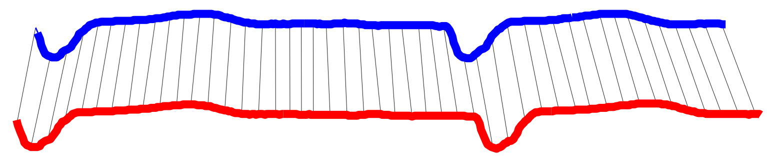
Uniform Scaling Technique



Euclidean Distance



Uniform Scaling Distance



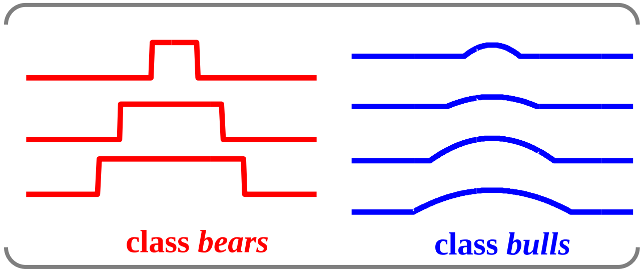
Using the *Euclidean* distance , the misalignment would cause a large error. However, the problem can be solved by using the *Uniform Scaling* distance.

The *Uniform Scaling* distance is a simple generalization of the *Euclidean* distance.

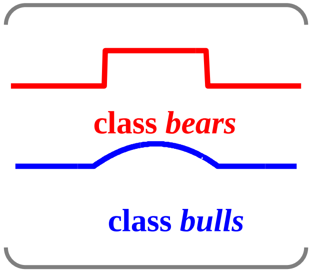
An Additional Insight on Data Redundancy

Uniform Scaling

✓to further reduce the size of data dictionary



left) **Data dictionary A**



right) **Data dictionary B**

✓to achieve lower error rate

Imagine the training data does contain some examples of gaits at speeds from 6.1 to 6.5km/h, unseen data contains 6.7km/h

Outline

- Motivation
- Proposed Framework
  - Concepts
  - Algorithms
- Experimental Evaluation
- Conclusion and Future Work

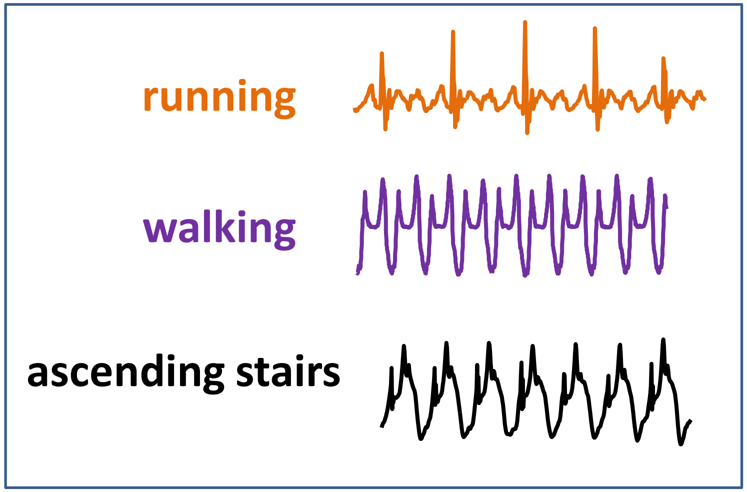
Classification using a Data Dictionary

Before showing how to build the data dictionary,
I want to show how to use it first.

Classification using a Data Dictionary

We use the classic one nearest neighbor algorithm

data dictionary



threshold

7.6

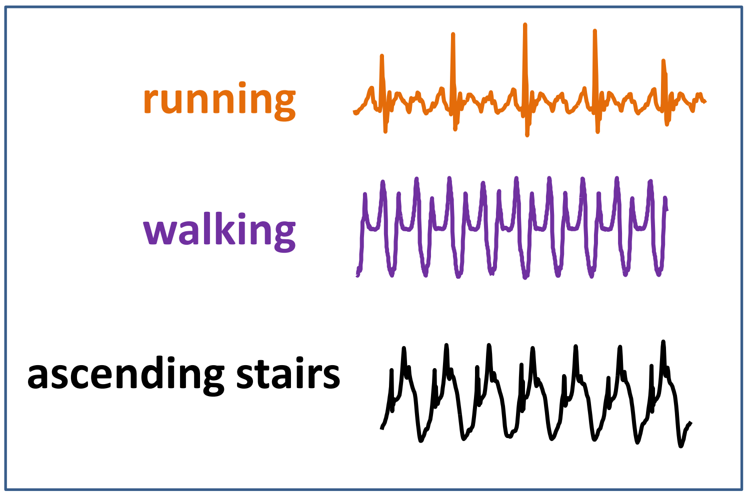
6.4

7.3

Classification using a Data Dictionary

We use the classic one nearest neighbor algorithm

data dictionary

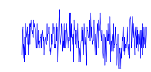


threshold

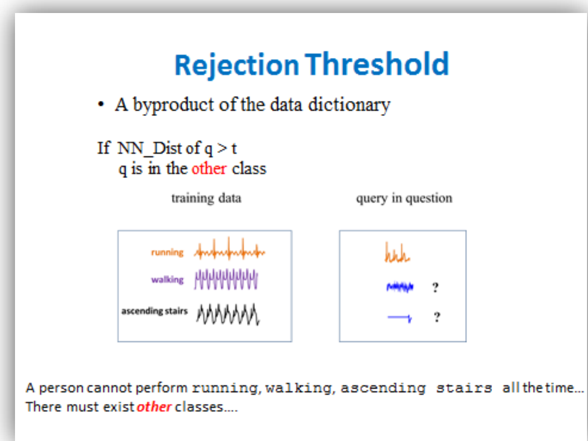
7.6

6.4

7.3

query : 

?



Building the Data Dictionary

Intuition

We show a toy dataset in the discrete domain to show the intuition.
Our goal remains large real-valued time series data

A ***weakly-labeled*** training dataset that contains two classes C1 and C2 :

C1 = { dpacekfjklwalkflwalkklpacedalyutekwalksfj}
C2 = { jhjhleapashljumpokdjklleaphfleapfjjumpacgd}

Building the Data Dictionary

Intuition

a training dataset that contains two classes C1 and C2 :

C1 = { dpacekfjklwalkflwalkklpacedalyutekwalksfj}
C2 = { jhjhleapashljumpokdjklleaphfleapfjjumpacgd}

- *weakly-labeled*
- the colored text is for introspection only

Building the Data Dictionary

Intuition

C1 = { dpacekfjklwalkflwalkklpacedalyutekwalksfj}
C2 = { jhjhleapashljumpokdjklleaphfleapfjjumpacgd}

data dictionary

C1: { **pace**, **walk** }
C2: { **leap** ; **jump**}

threshold

r = 1

Building the Data Dictionary

Intuition

data dictionary

C1: { **pace**, **walk** }
C2: { **leap** ; **jump**}

threshold

r = 1

Query :

ieap	NN_dist = 1	C2
kklp	NN_dist = 3	other

Building the Data Dictionary

Intuition

kklp dist = 3 `other`

What is the result if we do not have data dictionary ?

C1 = { dpacekfjklwalkflwalkklpacedalyutekwalksfj}

C2 = { jhjhleapashljumpokdjklleaphfleapfjjumpacgd}

kklp dist = 0 C1

Building the Data Dictionary

Intuition

Consider a streaming data that needs to be classified:

.. ttgpacedgrteweerjumpwalkflqrafertwqhafhfahfahfbseew..

How we build the data dictionary ?

Collecting statistics about which substrings are often used for correct prediction

Building the Data Dictionary

High-level Intuition

- To use a ***ranking function*** to score every subsequence in C.

- These “scores” rate the subsequences by their ***expected utility*** for classification of future unseen data.

- We use these scores to guide a greedy search algorithm, which ***iteratively*** selects the ***best subsequence*** and places it in **D**.

Building the Data Dictionary

Algorithm

How do we know this utility?

We estimate the utility by cross validation

Three steps below

Building the Data Dictionary

Step 1. The algorithm *scores* the subsequences in C.

Procedure :

(1). randomly extracted a large number of queries
(2). cross-validation
(3). rank every point in C using the SimpleRank function[a]

$$rank(x) = \sum_j \begin{cases} 1, & if \quad class(x) = class(x_j) \\ -2/(num\_of\_class - 1), & if \quad class(x) \neq class(x_j) \\ 0, & other \end{cases}$$

[a]K.Ueno, X. Xi, E. Keogh and D.J.Lee, Anytime Classification Using the Nearest Neighbor Algorithm with Applications to Stream Mining, ICDM, 2006

Building the Data Dictionary

SimpleRank function[a]

	S_1	S_2
classification accuracy	70%	70%

- However, suppose that S_1 is also very **close to** many objects with *different* class labels (***enemies***).
- If S2 keeps a larger distance from its enemy class objects, S_2 is a much better choice for inclusion in **D**.

[a]K.Ueno, X. Xi, E. Keogh and D.J.Lee, Anytime Classification Using the Nearest Neighbor Algorithm with Applications to Stream Mining, ICDM, 2006

Building the Data Dictionary

SimpleRank function[a]

$$rank(x) = \sum_j \begin{cases} 1, & if \quad class(x) = class(x_j) \\ -2/(num\_of\_class - 1), & if \quad class(x) \neq class(x_j) \\ 0, & other \end{cases}$$

- The intuition behind this algorithm is to give every instance a rank according to its ***contribution*** to the classification

- Score function ***rewards*** the subsequence that return ***correct*** classification and ***penalize*** those return ***incorrect*** classification

[a]K.Ueno, X. Xi, E. Keogh and D.J.Lee, Anytime Classification Using the Nearest Neighbor Algorithm with Applications to Stream Mining, ICDM, 2006

Building the Data Dictionary

The iteration procedure:

Step 1. The algorithm *scores* the subsequences in C.

Step 2. The *highest* scoring subsequence is *extracted* and placed in **D**.

Step 3. We identify all the queries that are incorrectly classified by the current **D**. These incorrectly classified items are passed back to **Step 1** to re-score the subsequences in C.

Building the Data Dictionary

Step 1. The algorithm *scores* the subsequences in C.

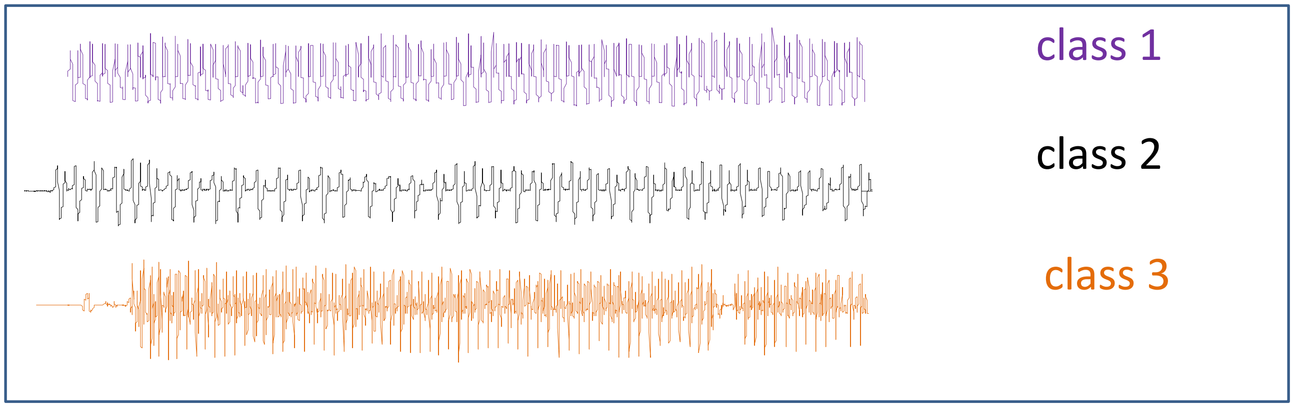
For simplicity, we use one query to illustrate how to score C.

We use one query to illustrate the ranking procedure **Step 1**

query q

?

weakly-labeled data

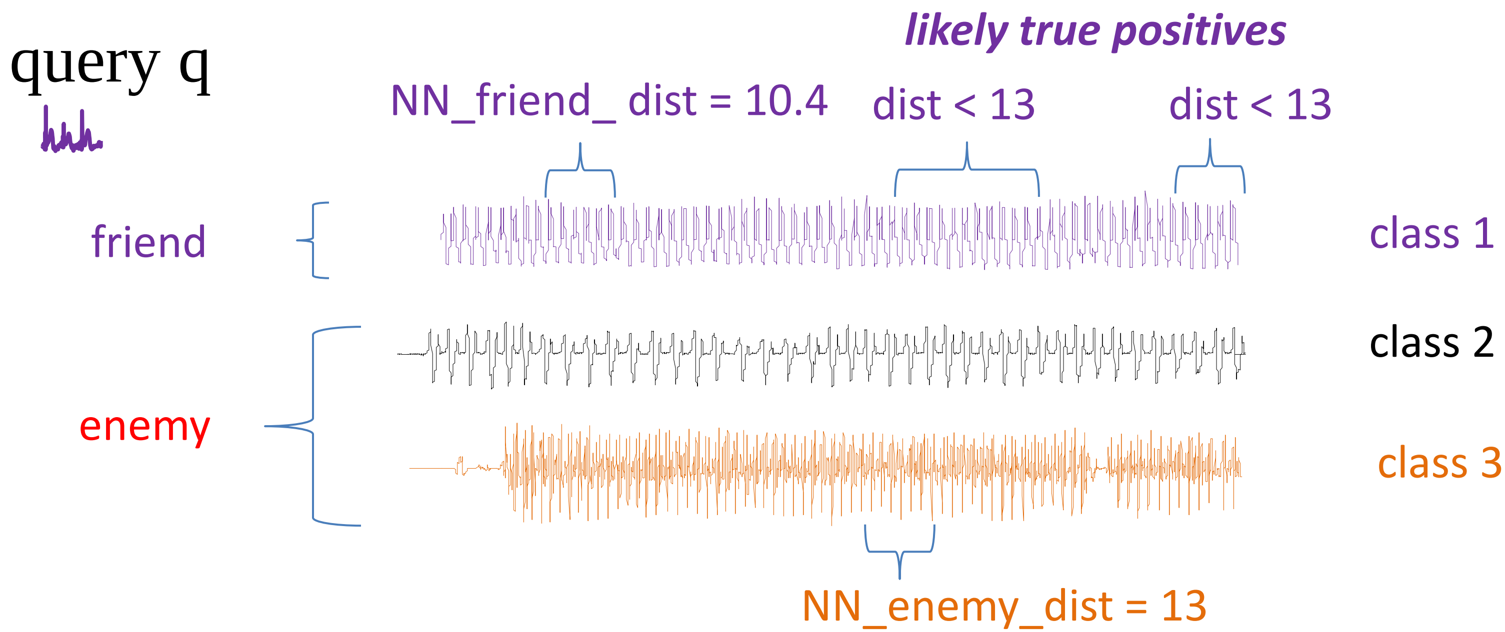


Perform one nearest neighbor classification

Two cases :

- when q is correctly classified
- when q is incorrectly classified

Step 1

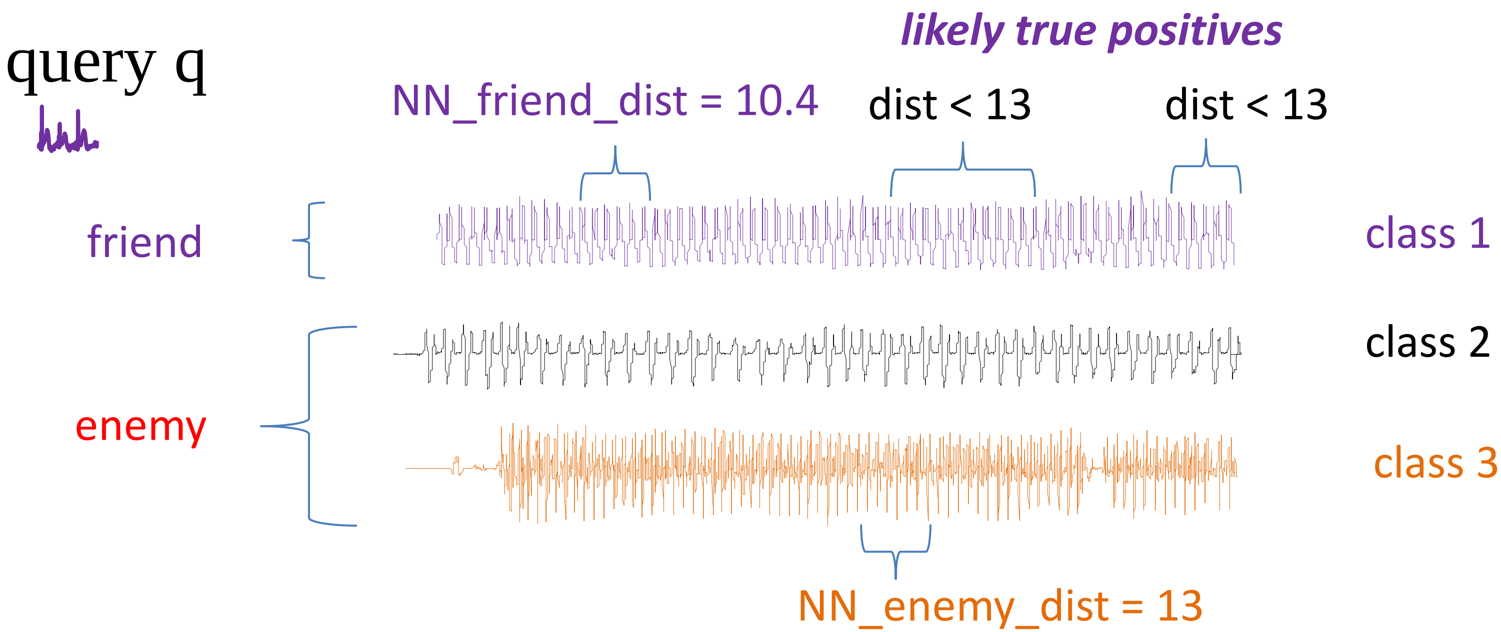


1. This query q is correctly classified as class 1
   NN_friend_dist = 10.4

2. found out the nearest neighbor distance in enemy (class 2 and class 3)is
   NN_enemy_dist = 13

3. For any subsequence that has nearest neighbor distance in friend class that is less than NN_enemy_dist , we give it a positive score.
   They are called ***nearest neighbor friends*** or ***likely true positives***

Step 1



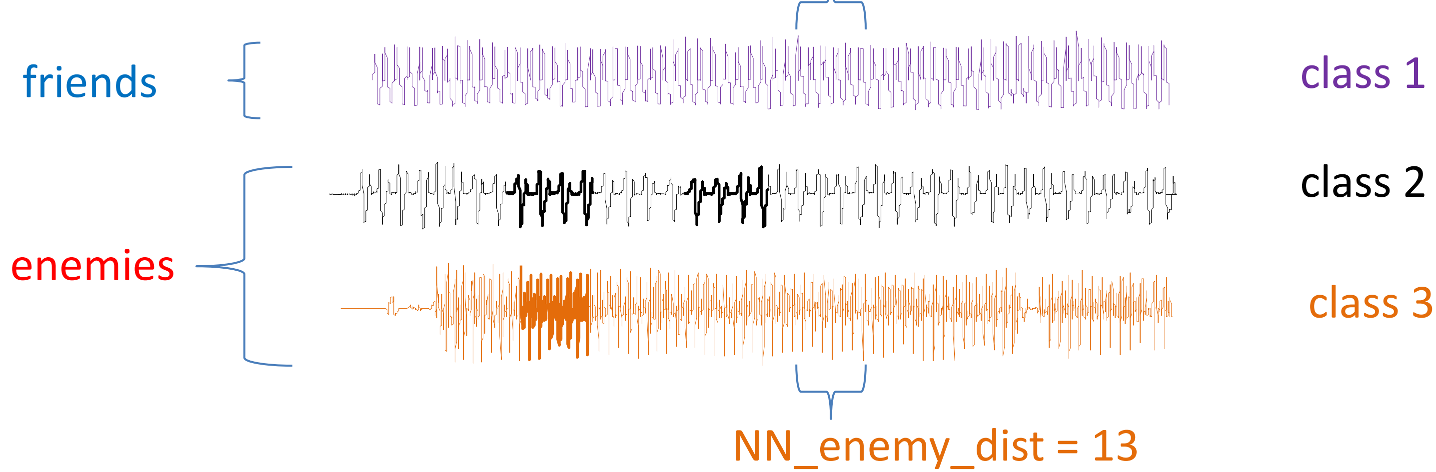
Two cases :

- If NN_friend_dist < NN_enemy_dist
  find ***nearest neighbor friends*** or ***likely true positives*** in the **friend** class

- If NN_friend_dist > NN_enemy_dist
  find ***nearest neighbor enemies*** or likely ***false positives*** in the enemy class

Step 1

query q

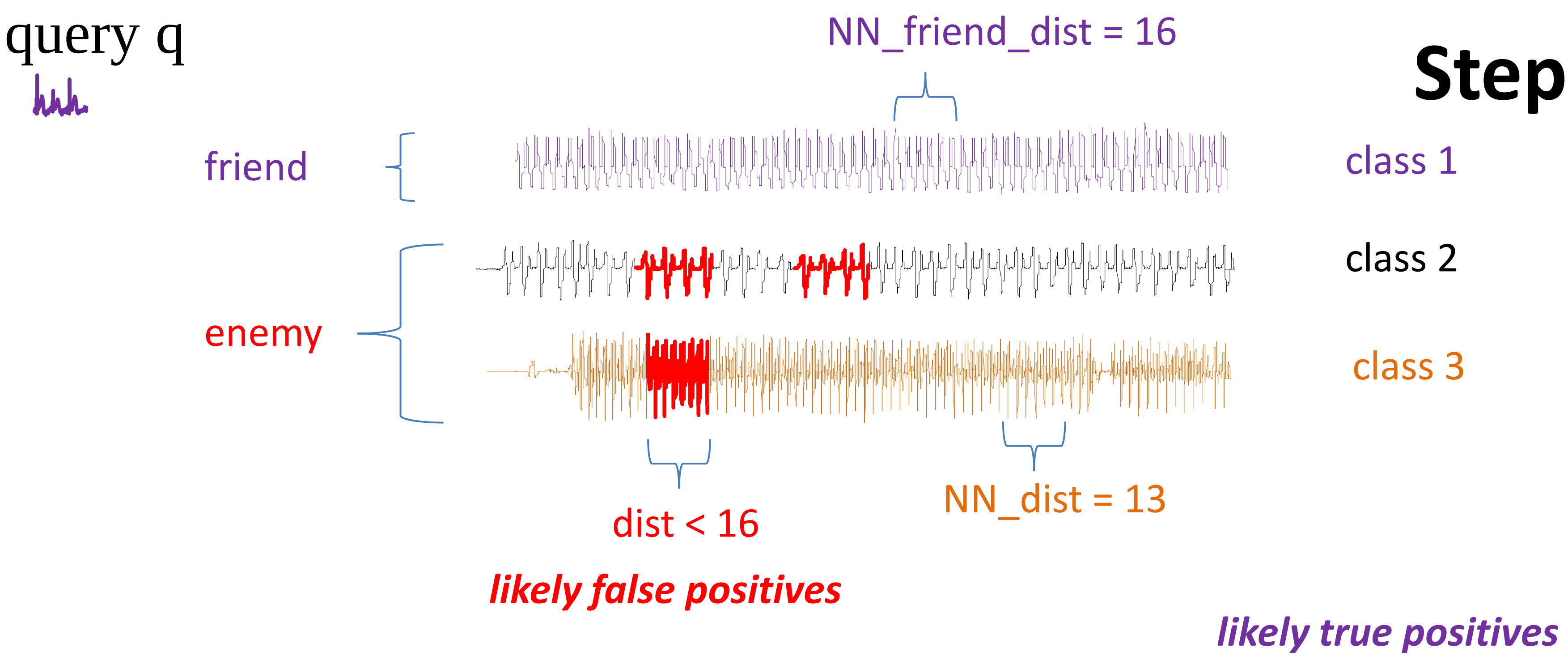


likely true positives

1. This query q is wrongly classified as class 3
   NN_enemy_dist = 13

2. found out the nearest neighbor distance in friends (class 1)
   NN_friend_dist = 16

Step 1

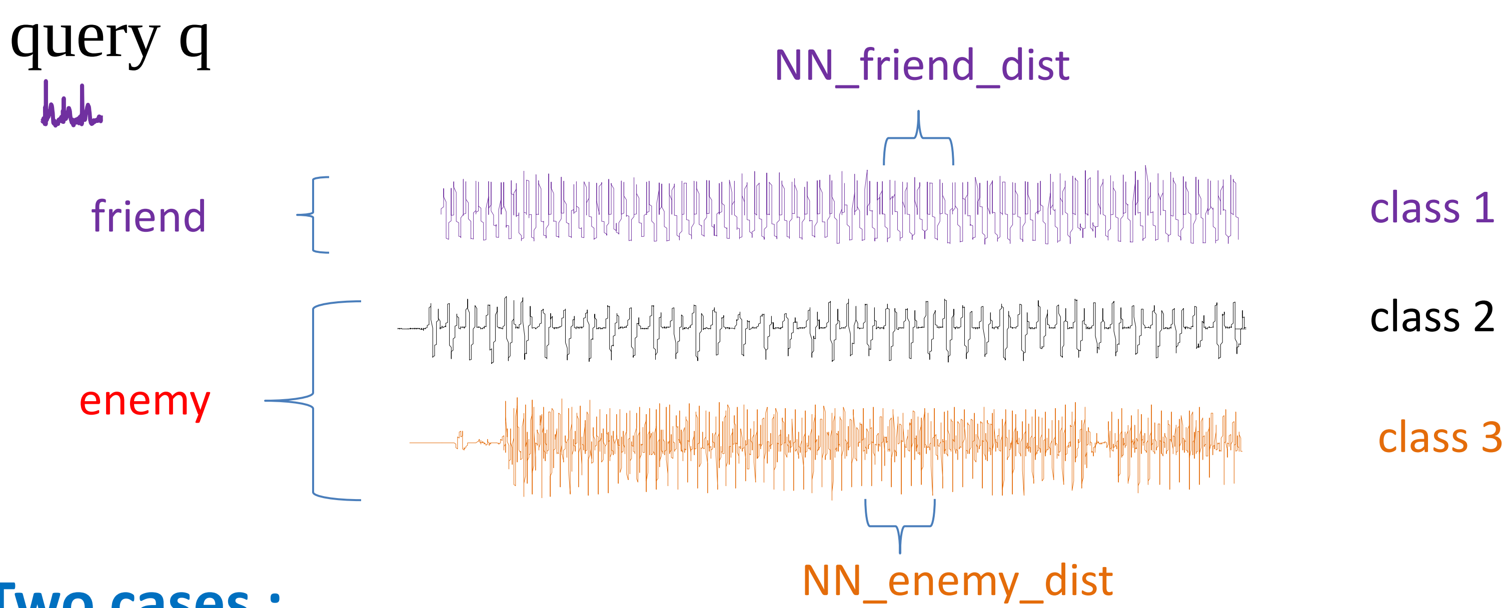


1. This query q is wrongly classified as class 3
   NN_enemy_dist = 13

2. found out the nearest neighbor distance in friend (class1)
   NN_friend_dist = 16

3. For any subsequence that has nearest neighbor distance in enemy class that is less than NN_friend_dist, we give it a negative score.
   They are called ***nearest neighbor enemies*** or ***likely false positives***

Step 1



Two cases :

If NN_friend_dist < NN_enemy_dist
find ***nearest neighbor friends*** or ***likely true positives*** in the **friend** class

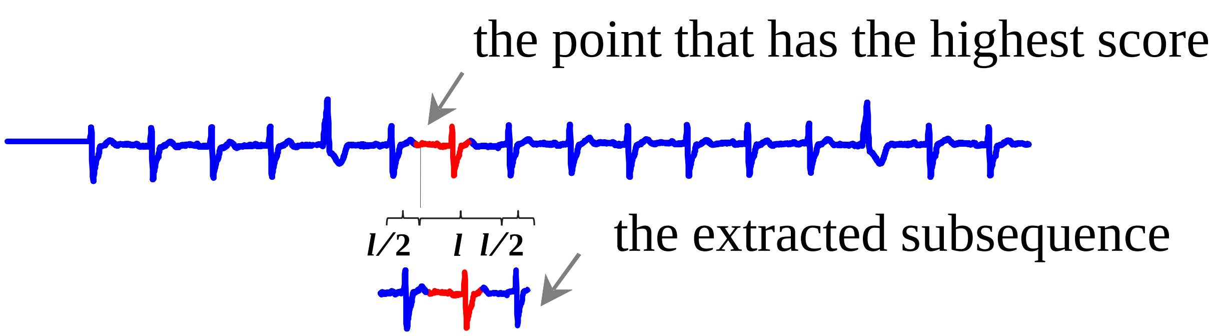
If NN_friend_dist > NN_enemy_dist
find ***nearest neighbor enemies*** or likely ***false positives*** in the enemy class

$$rank(S) = \sum_k \begin{cases} 1, & likely\ \ true\ \ positives \\ -2/(num\_of\_class-1), & likely\ \ false\ positives \\ 0, & other \end{cases}$$

Building the Data Dictionary

Step 2

The *highest* scoring subsequence is *extracted* and placed in **D**.



Building the Data Dictionary

Step 3

(1).Perform classification for all the queries using **D**.
(2).The incorrectly classified items are passed back to **Step 1** to re-score the subsequences in C.

Building the Data Dictionary

When to stop the iteration ?

- The accuracy of classification using just the data dictionary cannot be improved any more
- The size of the data dictionary

Building the Data Dictionary

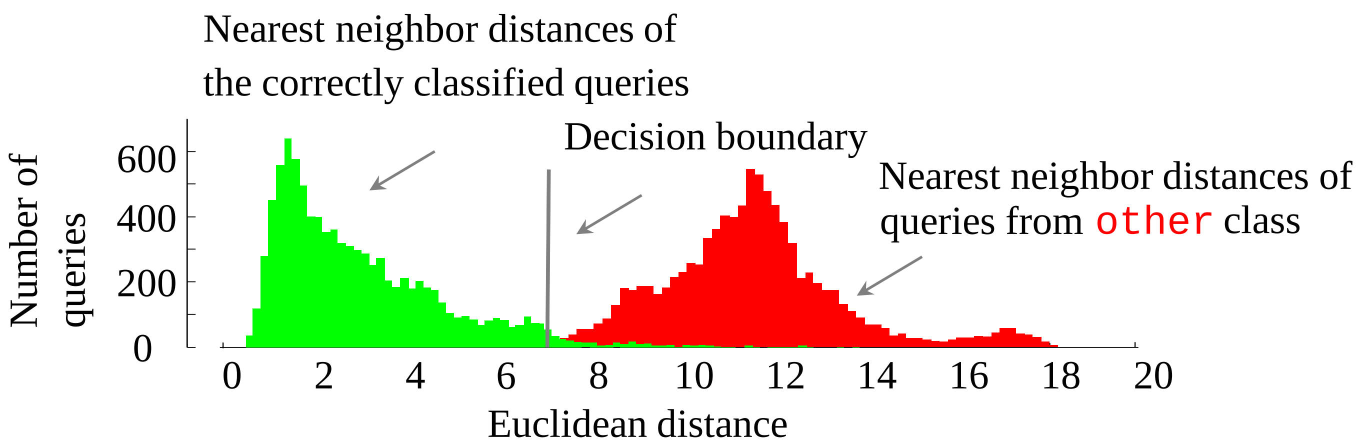
Learning the threshold distance

After the data dictionary is built, we learn a threshold to reject future queries, which do not belong to any of the learned classes.

Building the Data Dictionary

Learning the threshold distance

1. Record a **histogram** of the nearest neighbor distances of testing queries that are ***correctly*** classified using **D**
2. Record a **histogram** of the nearest neighbor distances of the queries in `other` classes



Uniform Scaling Technique

We replace the ***Euclidean*** distance with ***Uniform Scaling*** distance in the above data dictionary building and threshold learning process

Outline

- Motivation
- Proposed Framework
  - Concepts
  - Algorithms
- **Experimental Evaluation**
- Conclusion and Future Work

Experimental Evaluation

An Example Application in Physiology

Eight hours of data sampled at 110Hz was collected from wearable sensors on eight subjects' wrist, chest and shoes.

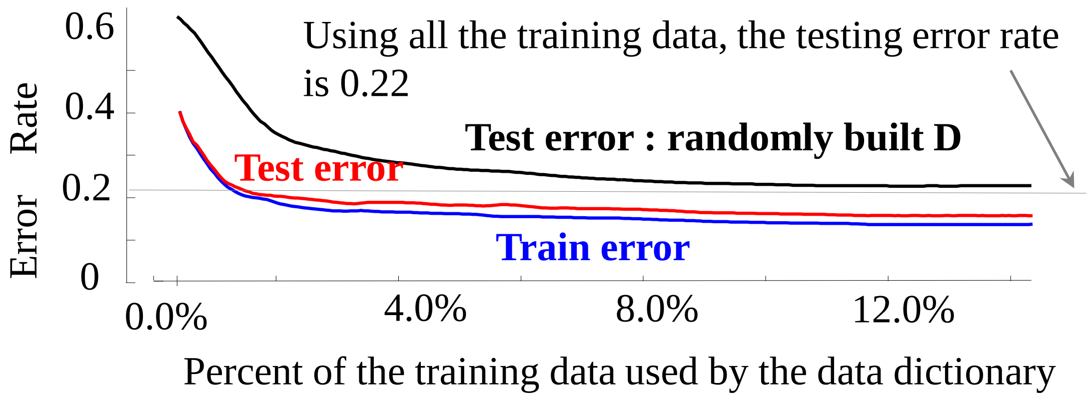
The activities includes :

`normal-walking`, `walking-very-slow`, `running`, `ascending-stairs`, `descending-stairs`, `cycling`,etc.

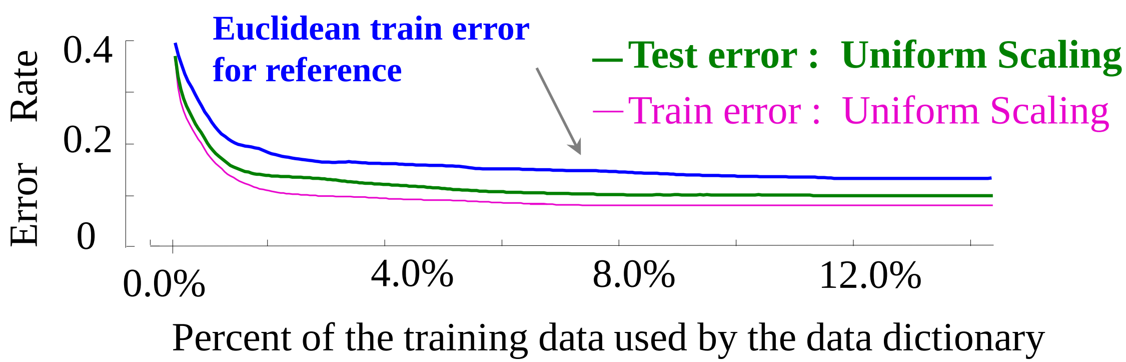
Experimental Evaluation

An Example Application in Physiology

Euclidean distance



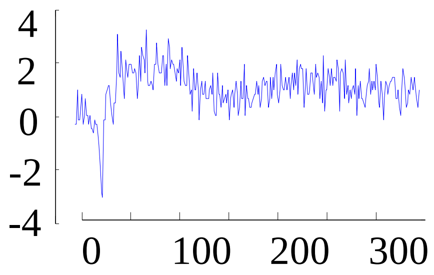
Uniform Scaling distance

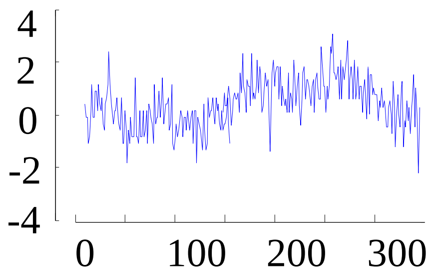


Experimental Evaluation

An Example Application in Physiology

Two examples of the rejected queries





Both queries contain significant amount of noise

Experimental Evaluation

An Example Application in Physiology

Rival Method

- We compare with the widely-used approach, which extracts signal features from the sliding windows. For fairness to this method, we used their suggested window size.

- We tested all the following classifiers : K-nearest neighbors, SVM, Naïve Bayes, Boosted decision trees, C4.5 decision tree

Experimental Evaluation

An Example Application in Physiology

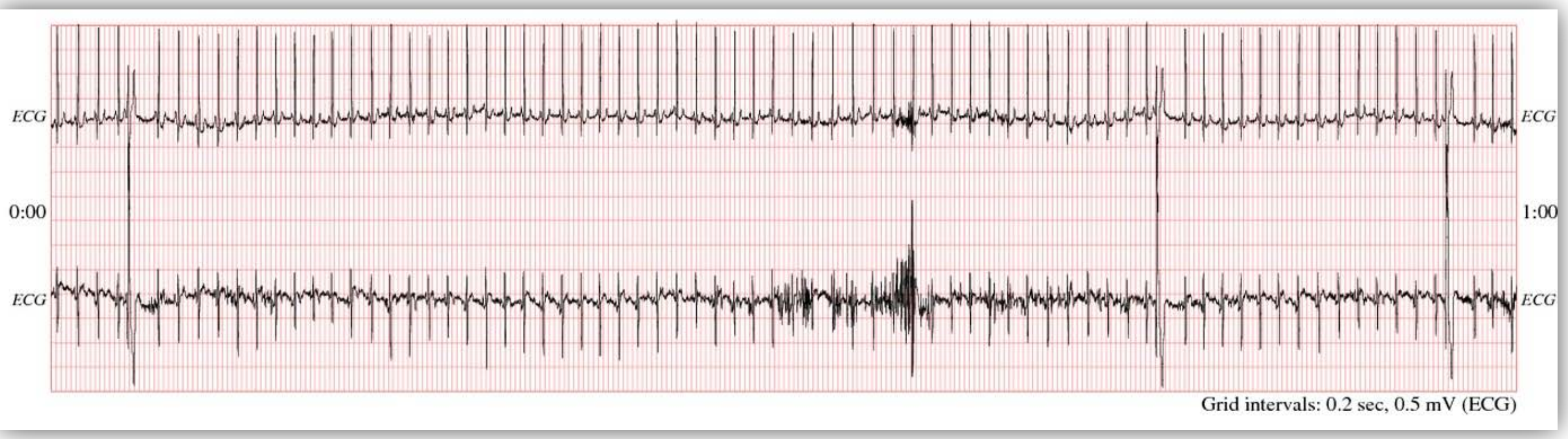
	Rival approach	Strawman	Our approach
error rate	0.364	0.221	0.152
amount of data used for classification	100%	100%	8.3%
assumptions	(1),(2),(3)	(1),(2),(3)	no assumption
running time	13 hours	28 hours	2.2 hours
rejected data	0	0	9.5%

Experimental Evaluation

An Example Application in Cardiology

The dataset includes ECG recordings from **fifteen** subjects with severe congestive heart failure.

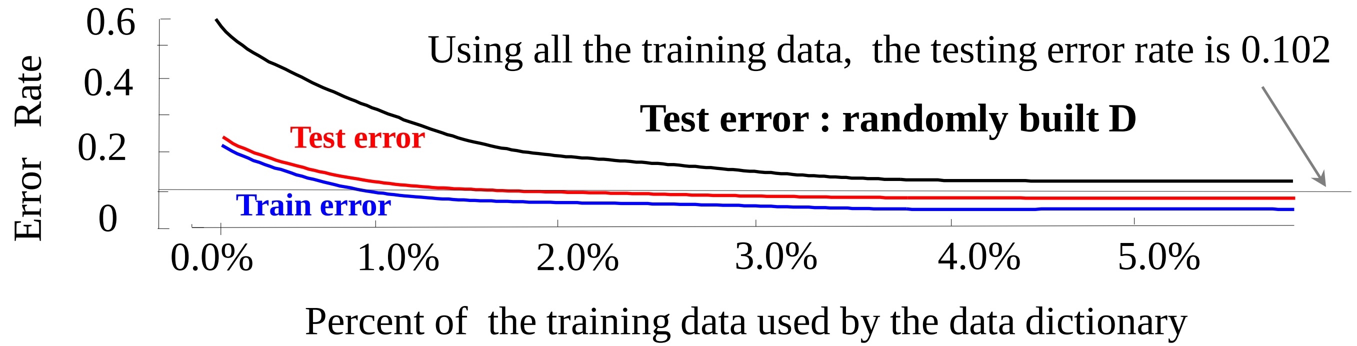
The individual recordings are each about **20 hours** in duration, samples at 250Hz



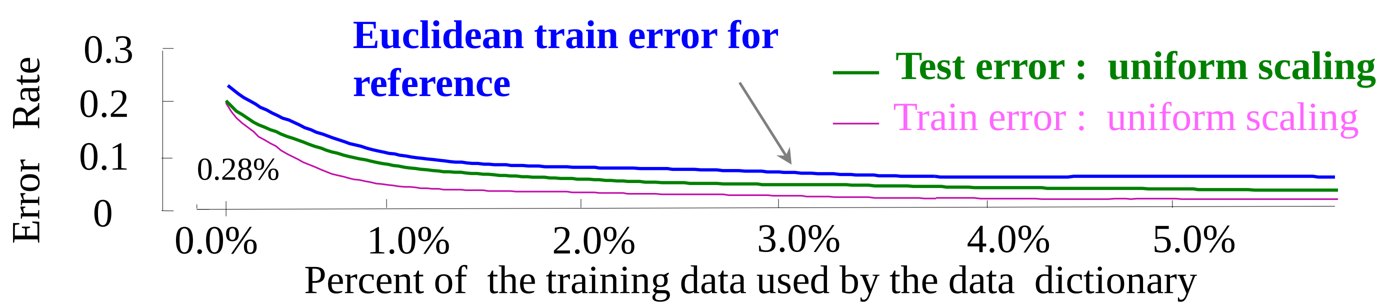
Experimental Evaluation

An Example Application in Cardiology

Euclidean distance



Uniform Scaling distance



Experimental Evaluation

An Example Application in Cardiology

	Rival approach	Strawman	Our approach
error rate	0.267	0.102	0.076
amount of data used for classification	100%	100%	2.1%
assumptions	(1),(2),(3)	(1),(2),(3)	no assumption
running time	78 hours	180 hours	3.6 hours
rejected data	0	0	4.8%

Experimental Evaluation

An Example Application in Daily Activities

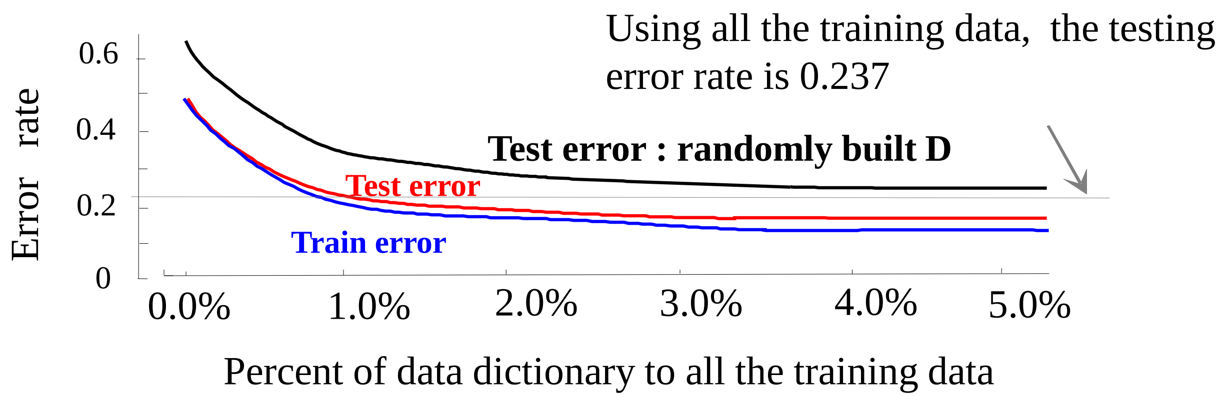
The MIT benchmark dataset that contains 20 subjects performing approximately 30 hours of daily activities.

such as: `running`, `stretching`, `scrubbing`, `vacuuming`, `riding-escalator`, `brushing-teeth`, `walking`, `bicycling`, etc. The data was sampled at 70 Hz.

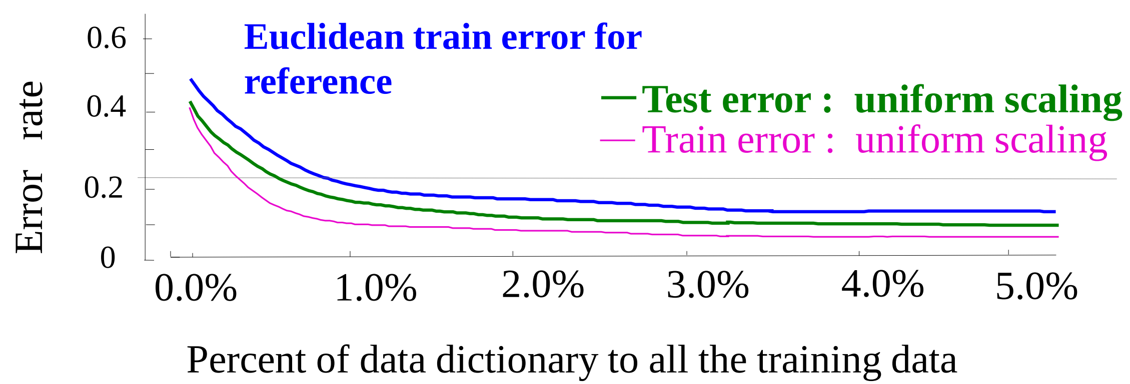
Experimental Evaluation

An Example Application in Daily Activities

Euclidean distance



Uniform Scaling distance



Experimental Evaluation

An Example Application in Daily Activities

	Rival approach	Strawman	Our approach
error rate	0.314	0.237	0.152
amount of data used for classification	100%	100%	3.8%
assumptions	(1),(2),(3)	(1),(2),(3)	no assumption
running time	52 hours	123 hours	4.8 hours
rejected	0	0	6.3%

Outline

- Motivation
- Proposed Framework
  - Concepts
  - Algorithms
- Experimental Evaluation
- **Conclusion and Future Work**

Conclusion

- Much of the progress in time series classification from streams in the last decade is almost ***Certainly Optimistic***
- Removing those unrealistic assumptions, we achieve much ***higher accuracy*** in ***a fraction*** of time

Conclusion

- Our approach requires only ***very weakly-labeled*** *data,* such as *“in this ten minutes of data, we see mostly normal heartbeats…..”,* **removing assumption (1)**

- Using this data we automatically build a “***data dictionary***”, which contains only the ***minimal subset*** of the original data to span the concept space. **This mitigates assumption (2)**

- As a byproduct of building this data dictionary, we learn a ***rejection threshold***, which allows us to **remove assumption (3)**

Thank you for your attention !

If you have any questions, please email bhu002@ucr.edu