DATA MINING 2
Time Series Classification

Riccardo Guidotti

a.a. 2020/2021

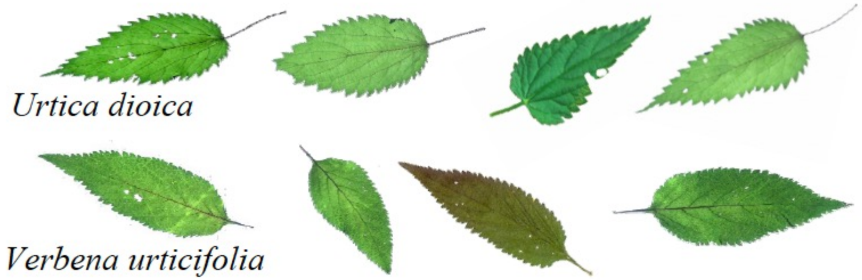
UNIVERSITÀ DI PISA

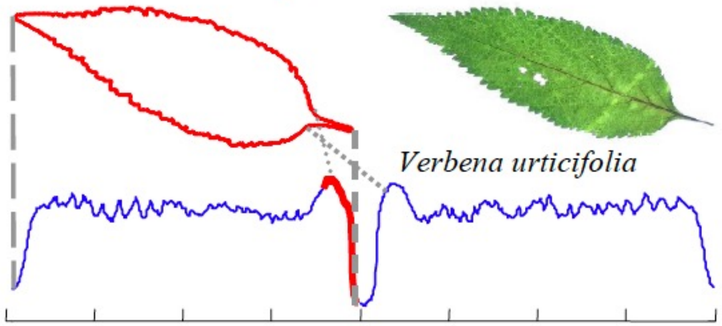
Time Series Classification

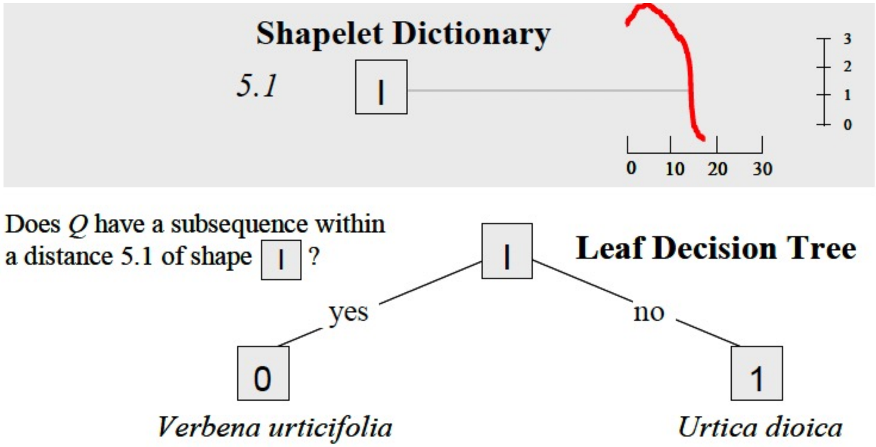
- Given a set X of n time series, $X = \{x_1, x_2, \ldots, x_n\}$, each time series has m ordered values $x_i = < x_{t1}, x_{t2}, \ldots, x_{tm} >$ and a class value c_i.
- The objective is to find a function f that maps from the space of possible time series to the space of possible class values.
- Generally, it is assumed that all the TS have the same length m.

Shapelet-based Classification

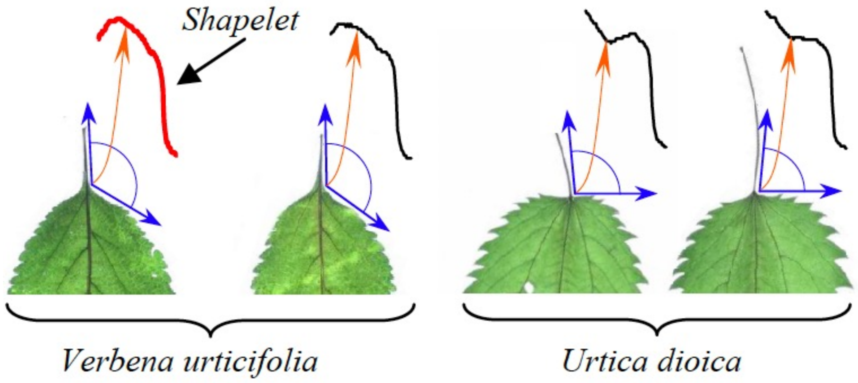
1. Represent a TS as a vector of distances with representative subsequences, namely shapelets.
2. Use it to as input for machine learning classifiers.



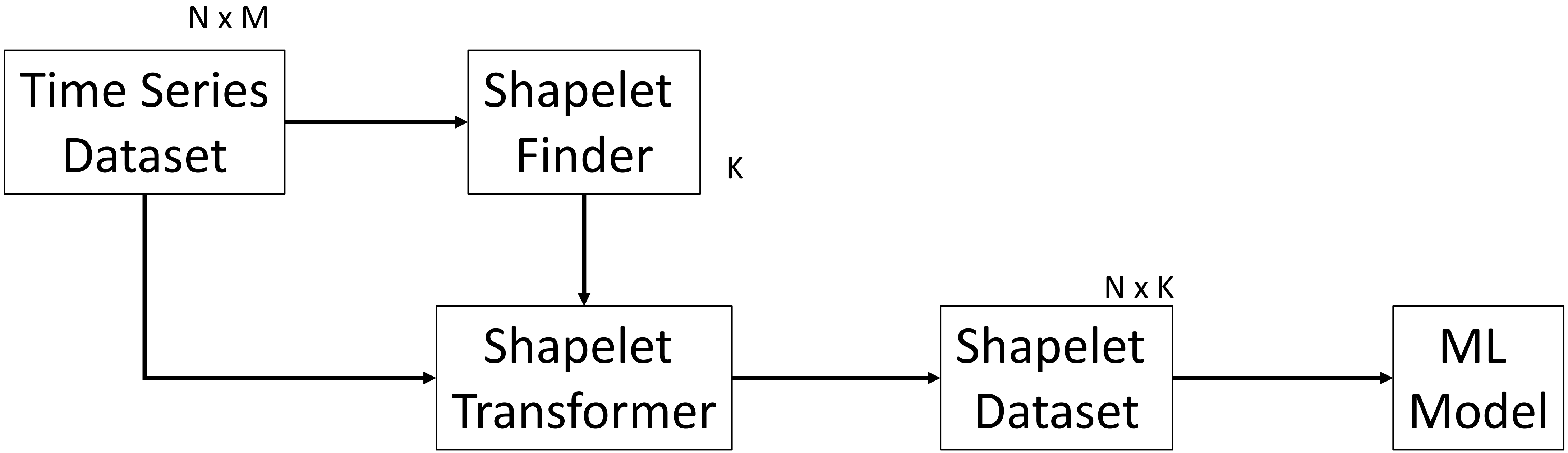




3.2	8.7
1.4	7.9
6.7	4.2
9.2	3.4



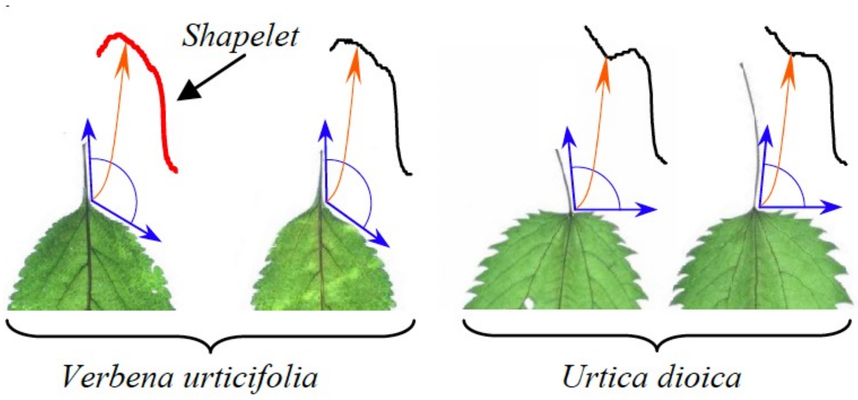
Shaplet-based Classifiers



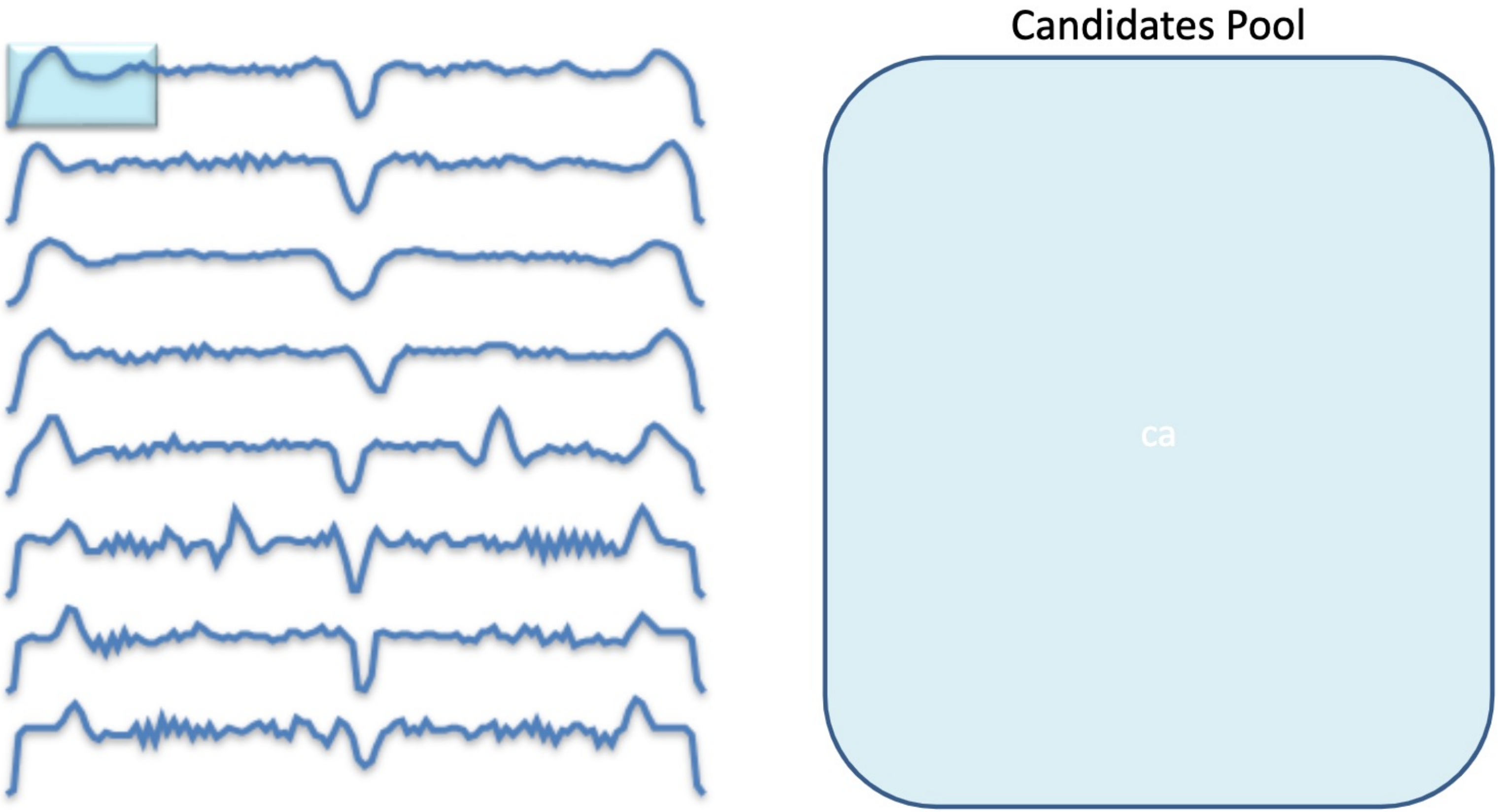
Time Series Shapelets

- Shapelets are TS subsequences which are maximally representative of a class.
- Shapelets can provide interpretable results, which may help domain practitioners better understand their data.
- Shapelets can be significantly more accurate/robust because they are *local features*, whereas most other state-of-the-art TS classifiers consider *global features*.

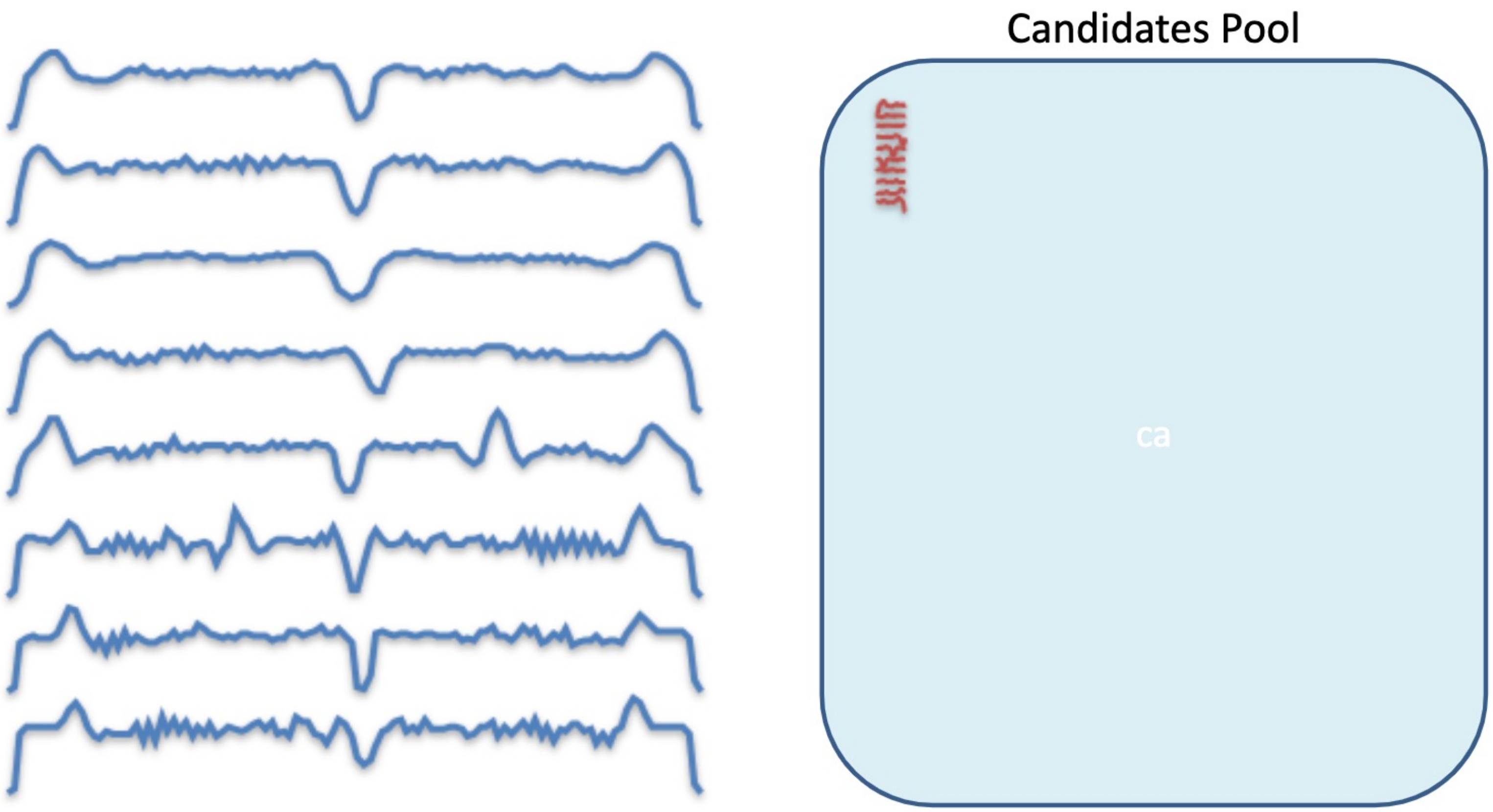
Verbena	0.87
Urtica	0.34



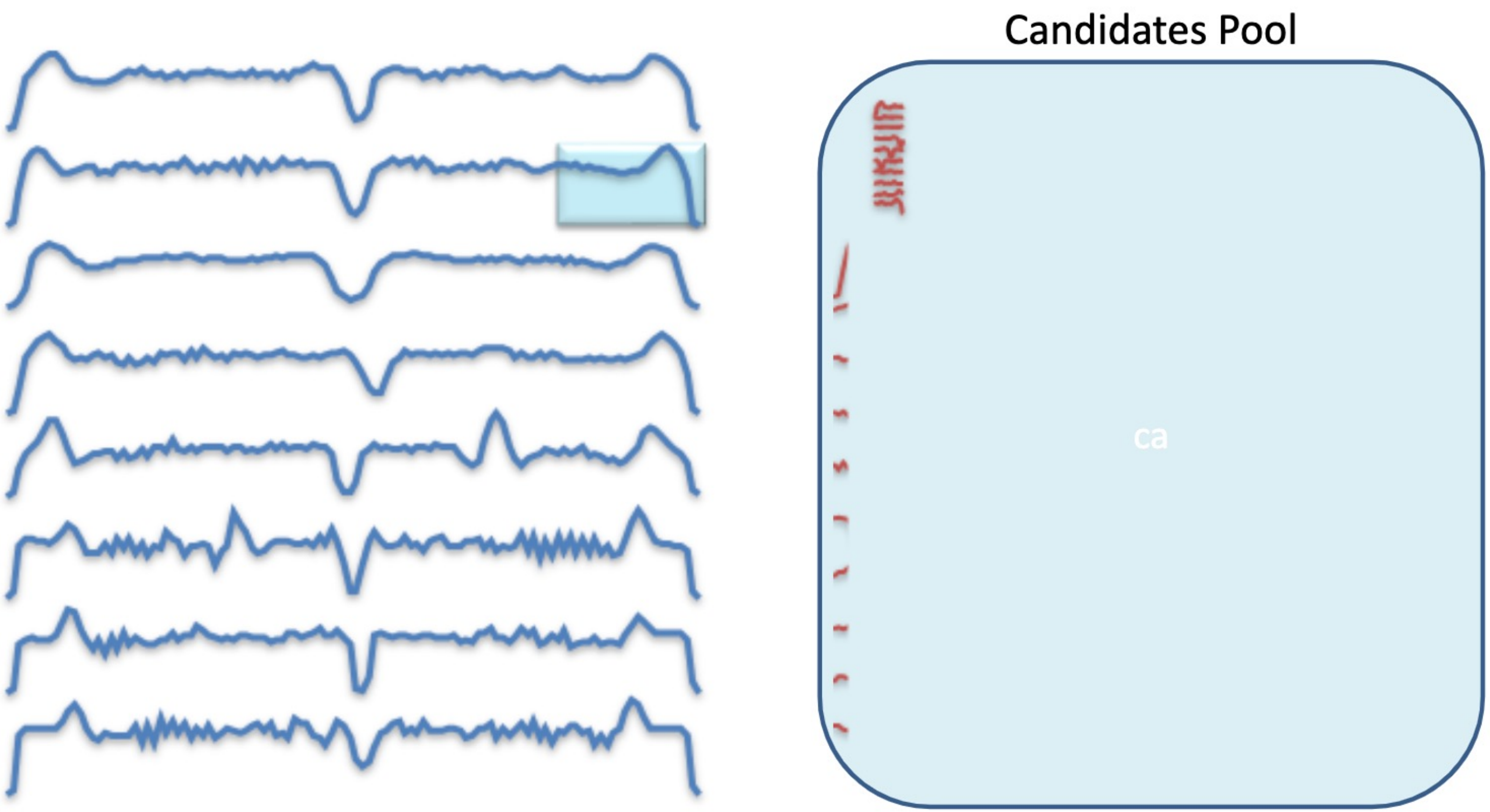
Extract Subsequences of all Possible Lengths



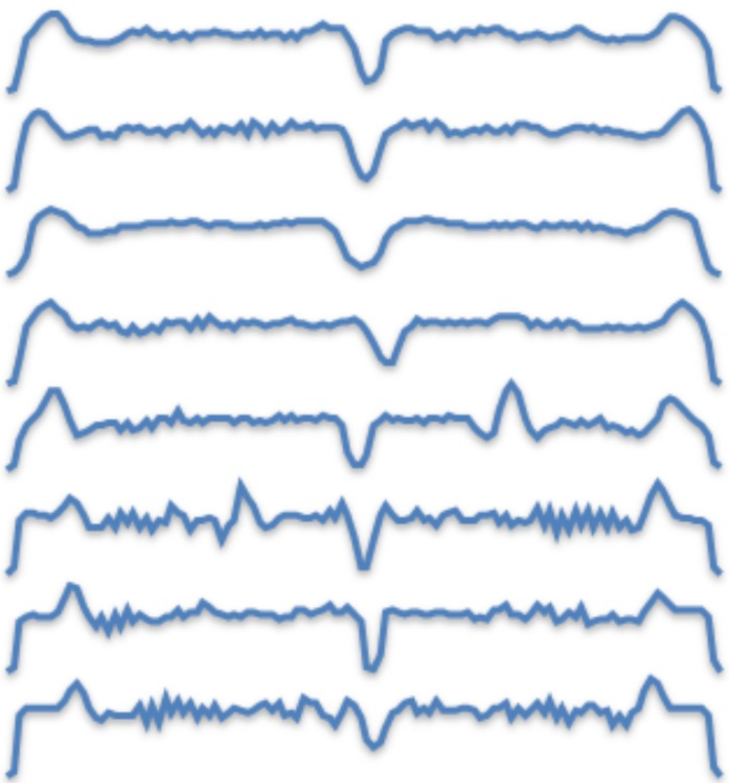
Extract Subsequences of all Possible Lengths

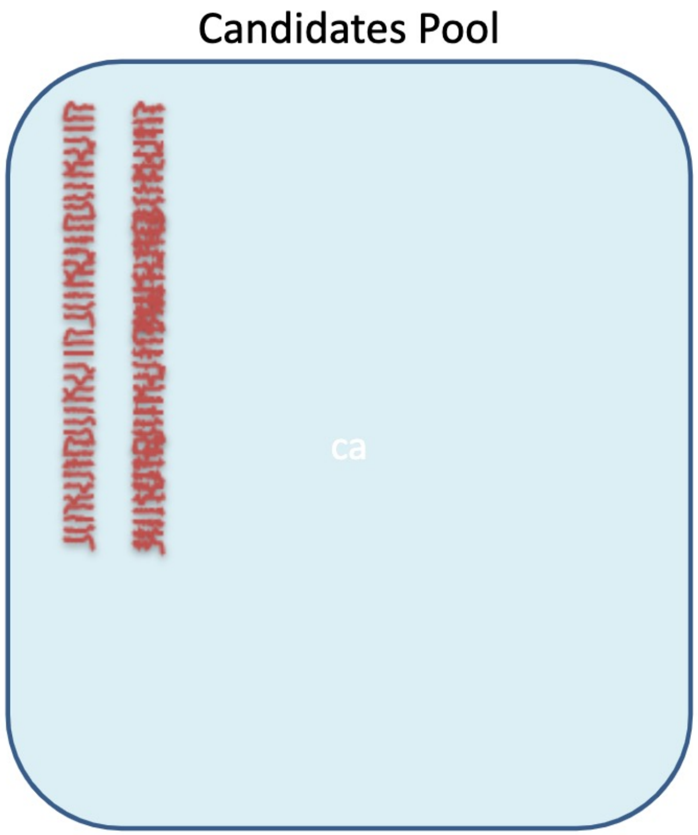


Extract Subsequences of all Possible Lengths

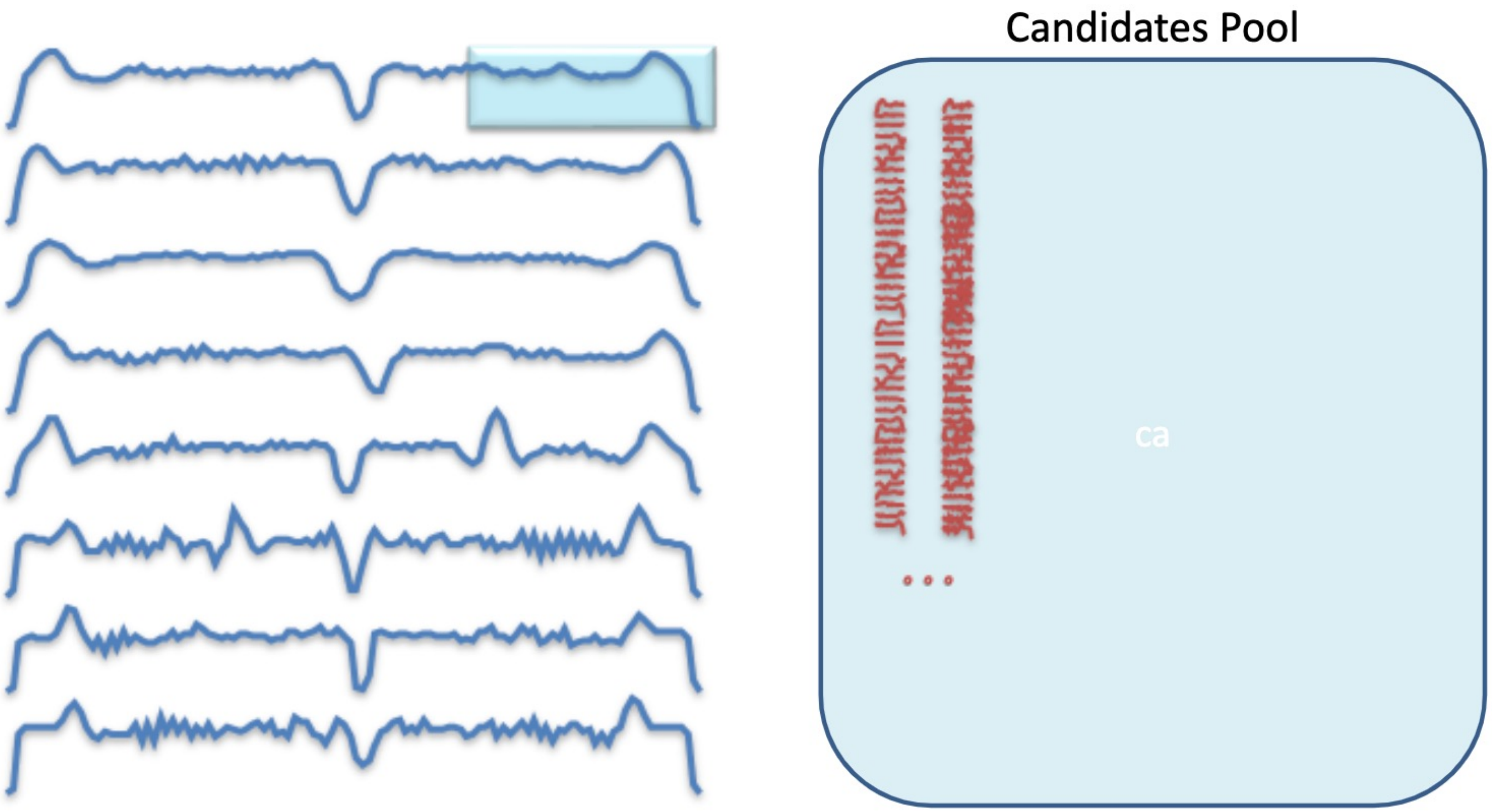


Extract Subsequences of all Possible Lengths



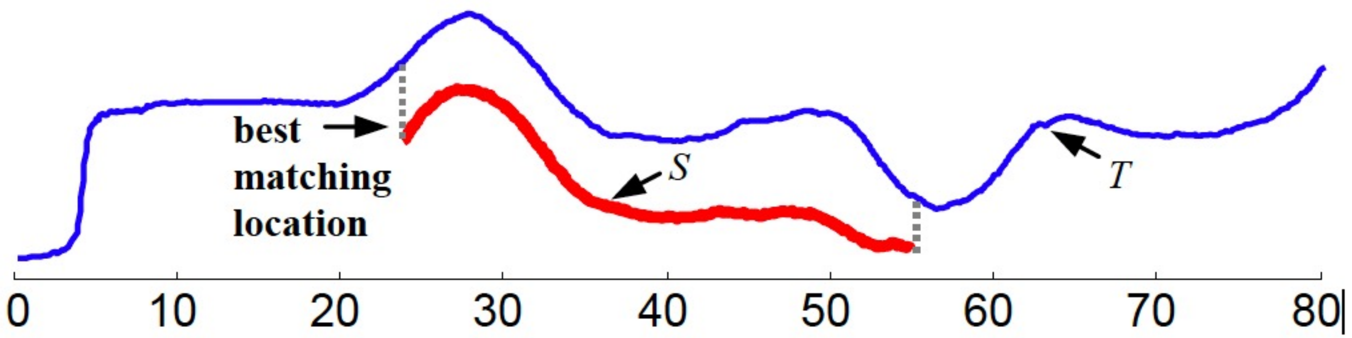


Extract Subsequences of all Possible Lengths



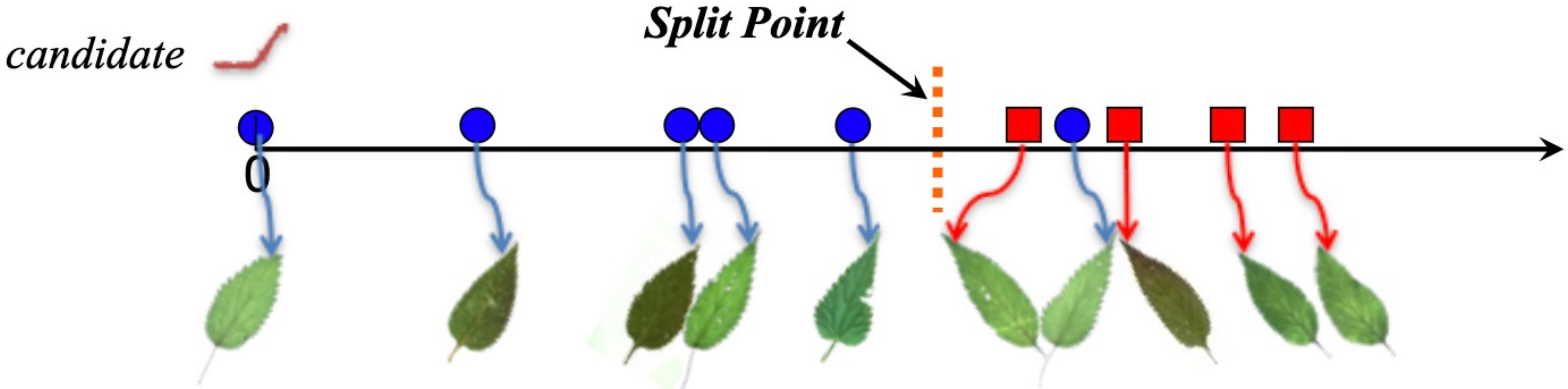
Distance with a Subsequence

- Distance from the TS to the subsequence *SubsequenceDist(T, S)* is a distance function that takes time series *T* and subsequence *S* as inputs and returns a nonnegative value *d*, which is the distance from *T* to *S*.
- $SubsequenceDist(T, S) = min(Dist(S, S')), for\ S' \in S_T^{|S|}$
- where $S_T^{|S|}$ is the set of all possible subsequences of *T*
- Intuitively, it is the distance between *S* and its best matching location in *T*.

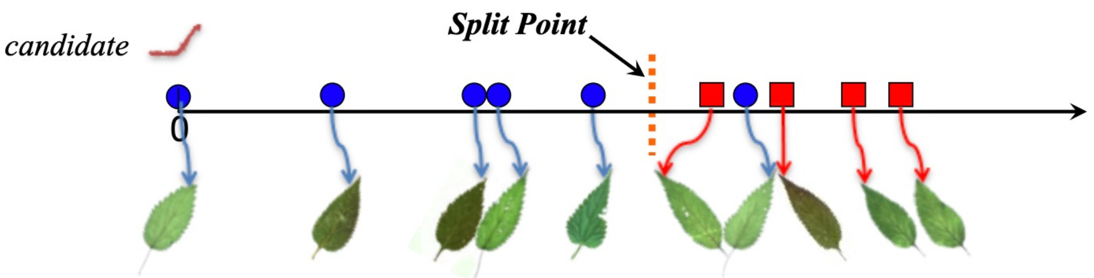


Testing The Utility of a Candidate Shapelet

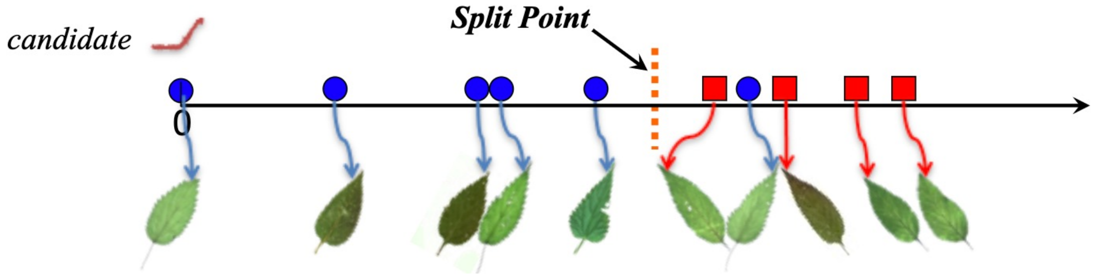
- Arrange the TSs in the dataset *D* based on the distance from the candidate.
- Find the optimal split point that maximizes the information gain (same as for Decision Tree classifiers)
- Pick the candidate achieving best utility as the shapelet



Entropy



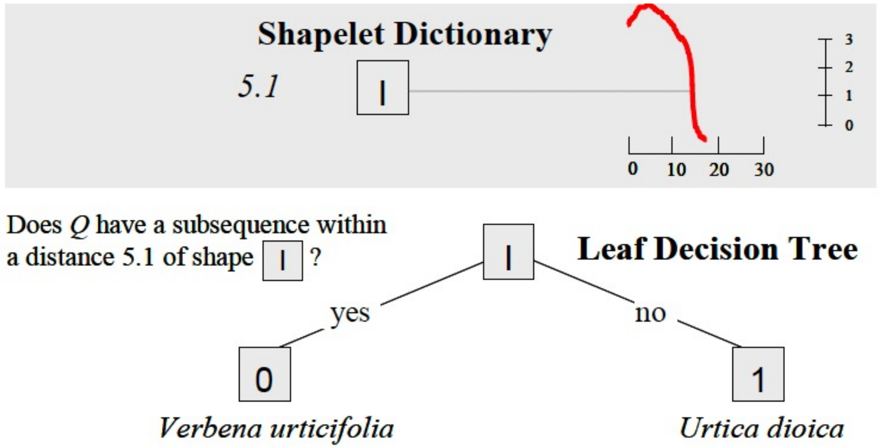
- A TS dataset D consists of two classes, A and B.
- Given that the proportion of objects in class A is $p(A)$ and the proportion of objects in class B is $p(B)$,
- The **Entropy** of D is: $I(D) = -p(A)log(p(A)) -p(B)log(p(B))$.
- Given a strategy that divides the D into two subsets D_1 and D_2, the information remaining in the dataset after splitting is defined by the weighted average entropy of each subset.
- If the fraction of objects in D_1 is $f(D_1)$ and in D_2 is $f(D_2)$,
- The total entropy of D after splitting is $\hat{I}(D) = f(D_1)I(D_1) + f(D_2)I(D_2)$.



Information Gain

Split point distance from shapelet = 5.1

- Given a certain split strategy *sp* which divides *D* into two subsets D_1 and D_2, the entropy before and after splitting is $I(D)$ and $\hat{I}(D)$.
- The **information gain** for this splitting rule is:
- $Gain(sp) = I(D) - \hat{I}(D) =$
- $= I(D) - f(D_1)I(D_1) + f(D_2)I(D_2)$.



- We use the distance from *T* to a shapelet *S* as the splitting rule *sp*.

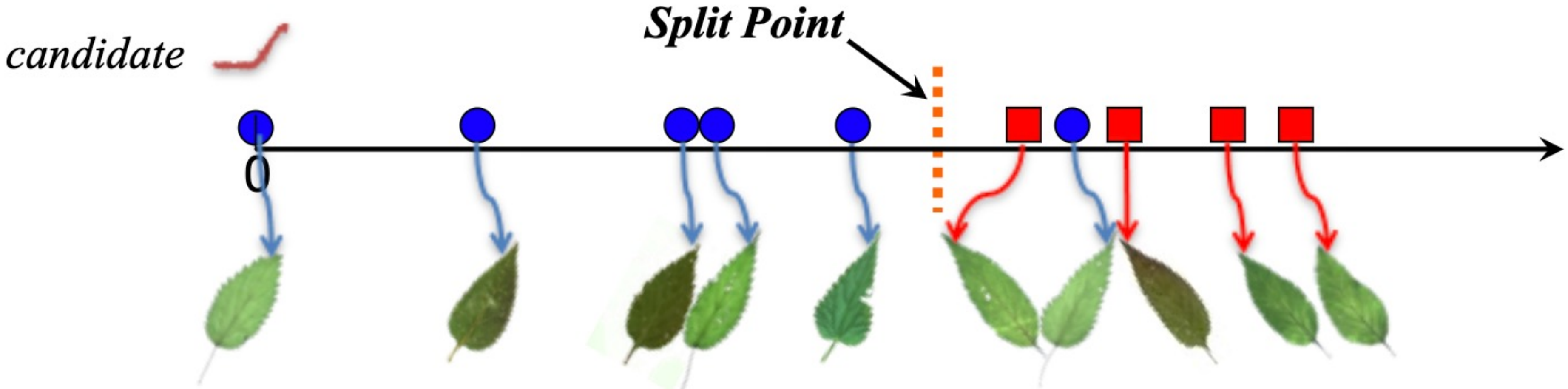
Problem

- The total number of candidate is $$\sum_{l=MINLEN}^{MAXLEN} \sum_{T_i \in D} (|T_i| - l + 1)$$

- For each candidate you have to compute the distance between this candidate and each training sample
- For instance
  - 200 instances with length 275
  - 7,480,200 shapelet candidates

Speedup

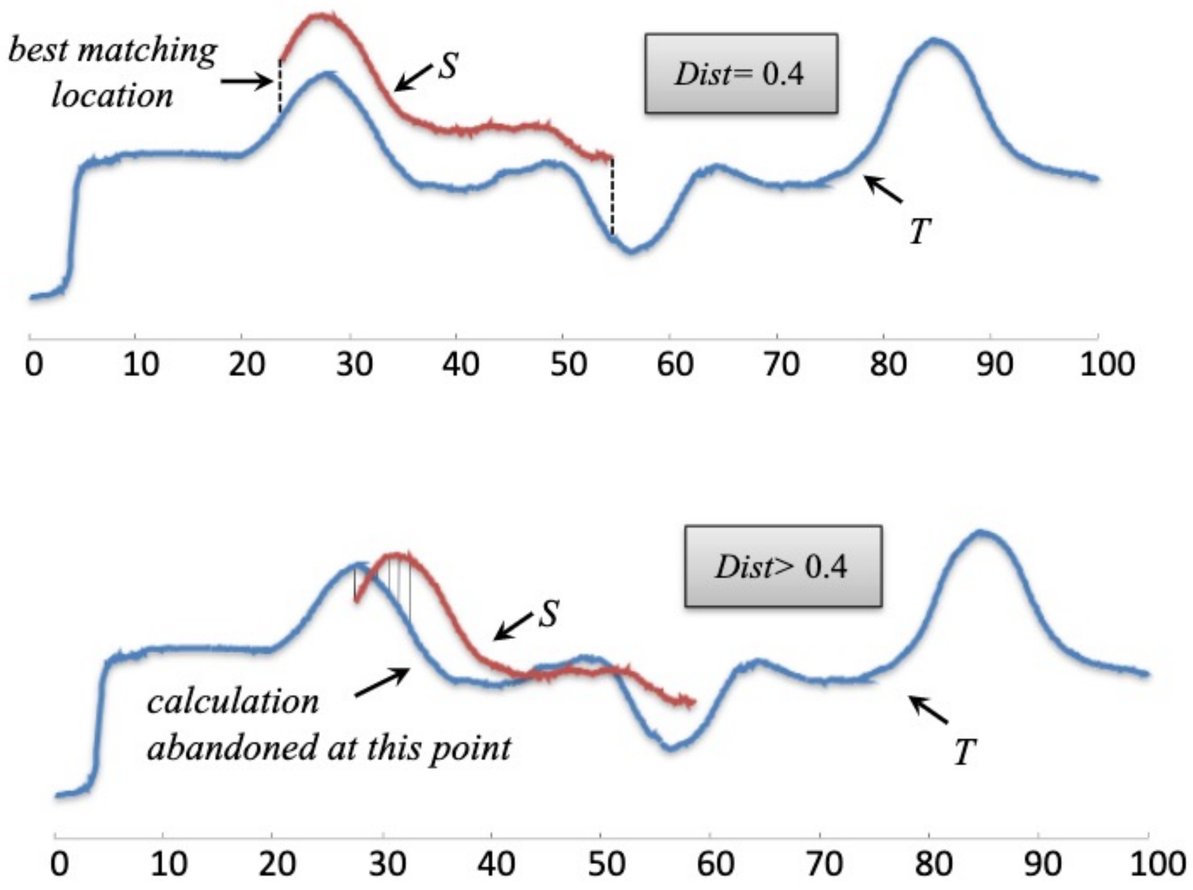
- Distance calculations form TSs to shapelet candidates is expensive.
- Reduce the time in two ways
- Distance Early Abandon
  - reduce the distance computation time between two TS
- Admissible Entropy Pruning
  - reduce the number of distance calculations



Distance Early Abandon

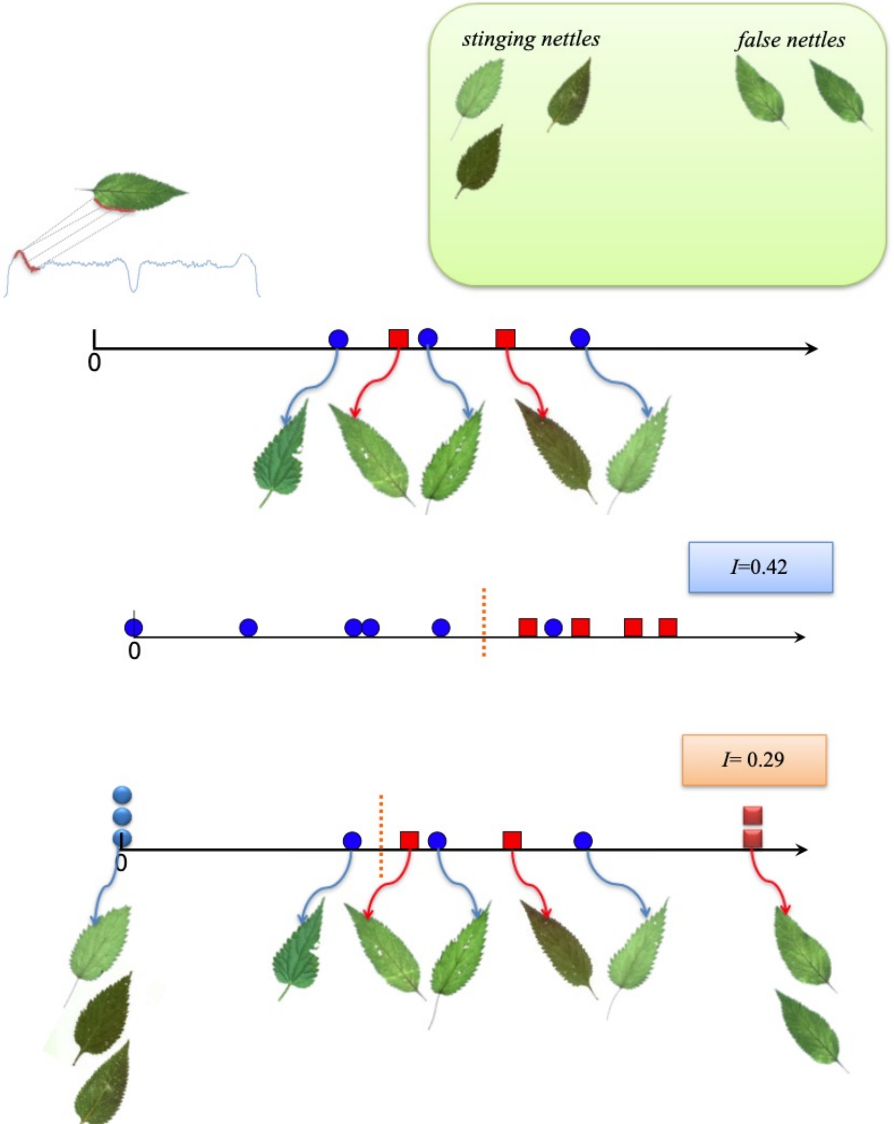
- We only need the minimum distance.

- Method
  - Keep the best-so-far distance
  - Abandon the calculation if the current distance is larger than best-so-far.



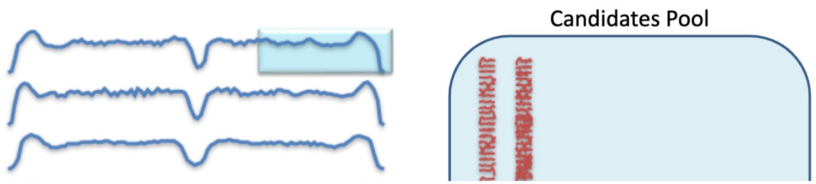
Admissible Entropy Pruning

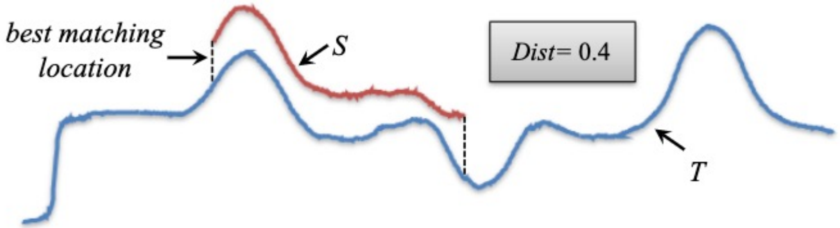
- We only need the best shapelet for each class
- For a candidate shapelet
  - We do not need to calculate the distance for each training sample
  - After calculating some training samples, the upper bound of information gain < best candidate shapelet
  - Stop calculation
  - Try next candidate

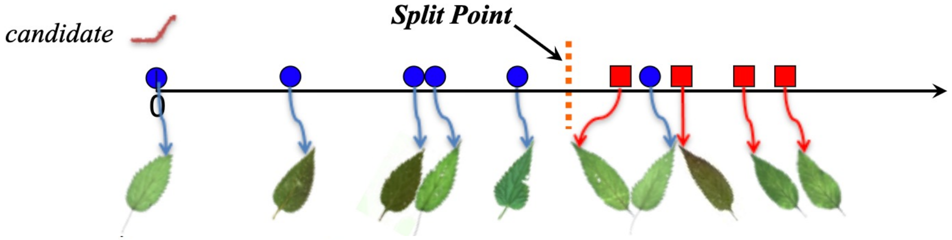


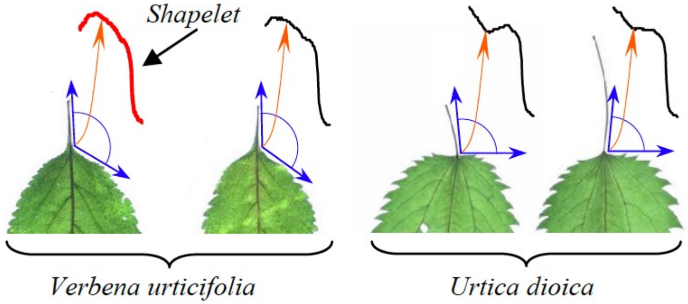
Shapelet Summary

1. Extract all possible subsequences of a set given lengths (candidate shapelets)
2. For each candidate shapelet
   1. Calculate the distance with each time series keeping the minimum distance (best alignment)
   2. Evaluate the discriminatory effect of the shapelet through the Information Gain
3. Return the *k* best shapelets with the highest Information Gain.
4. Transform a dataset and train a ML model.









An Alternative Way for Extracting Shapelets

- The minimum distances (M) between Ts and Shapelets can be used as predictors to approximate the TSs label (Y) using a linear model (W):

$$\hat{Y}_i = W_0 + \sum_{k=1}^{K} M_{i,k} W_k, \quad \forall i \in \{1, \ldots, I\}$$

- A logistic regression loss can measure the quality of the prediction:

$$\mathcal{L}(Y, \hat{Y}) = -Y \ln \sigma(\hat{Y}) - (1 - Y) \ln \left(1 - \sigma(\hat{Y})\right)$$

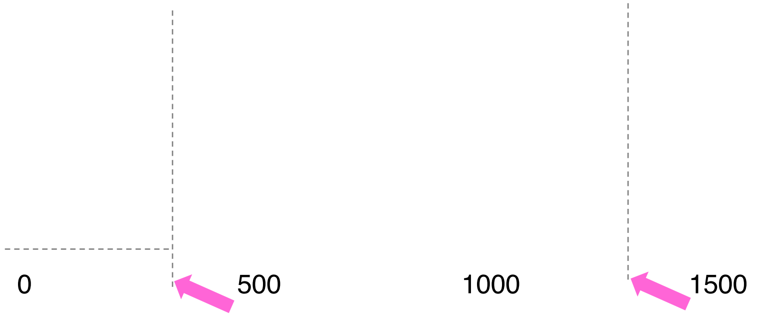
- The objective is to minimize a regularized loss function across all the instances (I) :

$$\underset{S,W}{\operatorname{argmin}} \ \mathcal{F}(S, W) = \underset{S,W}{\operatorname{argmin}} \sum_{i=1}^{I} \mathcal{L}(Y_i, \hat{Y}_i) + \lambda_W ||W||^2$$

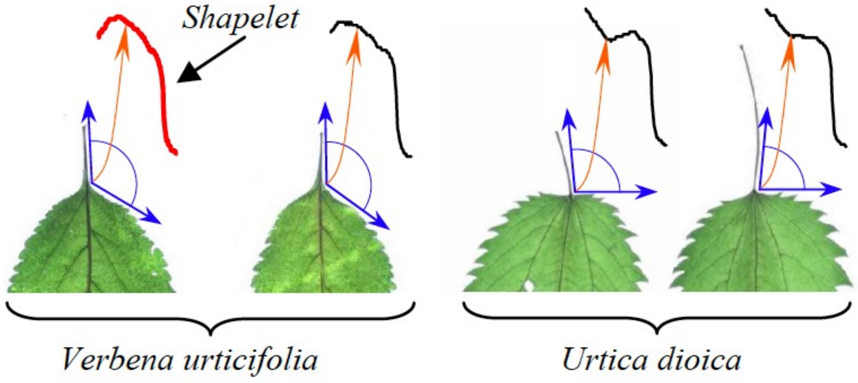
- We can find the optimal shapelet for the objective function in a NN fashion by updating the shapelets in the minimum direction of the objective, hence the first gradient. Similarly, the weights can be jointly updated towards minimizing the objective function.

Motif/Shapelet Summary

- A **motif** is a repeated pattern/subsequence in a given TS.



- A **shapelet** is a pattern/subsequence which is maximally representative of a class with respect to a given dataset of TSs.



References

- Matrix Profile I: All Pairs Similarity Joins for Time Series: A Unifying View that Includes Motifs, Discords and Shapelets. Chin-Chia Michael Yeh et al. 1997
- Time Series Shapelets: A New Primitive for Data Mining. Lexiang Ye and Eamonn Keogh. 2016.
- Josif Grabocka, Nicolas Schilling, Martin Wistuba, Lars Schmidt-Thieme (2014): Learning Time-Series Shapelets, in Proceedings of the 20th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2014
- Deep learning for time series classication: a review. Hassan Ismail Fawaz et al. 2019.

References

- Selective review of offline change point detection methods. Truong, C., Oudre, L., & Vayatis, N. (2020). Signal Processing, 167, 107299.
- Time Series Analysis and Its Applications. Robert H. Shumway and David S. Stoffer. 4th edition.(https://www.stat.pitt.edu/stoffer/tsa4/tsa4.pdf)
- Mining Time Series Data. Chotirat Ann Ratanamahatana et al. 2010. (https://www.researchgate.net/publication/227001229_Mining_Time_Series_Data)
- Dynamic Programming Algorithm Optimization for Spoken Word Recognition. Hiroaki Sakode et al. 1978.
- Experiencing SAX: a Novel Symbolic Representation of Time Series. Jessica Line et al. 2009
- Compression-based data mining of sequential data. Eamonn Keogh et al. 2007.