Time Series Management

Michele Linardi Ph.D.

michele.linardi@orange.fr

Syllabus

- Time series classification
- Shapelet Discovery primitive

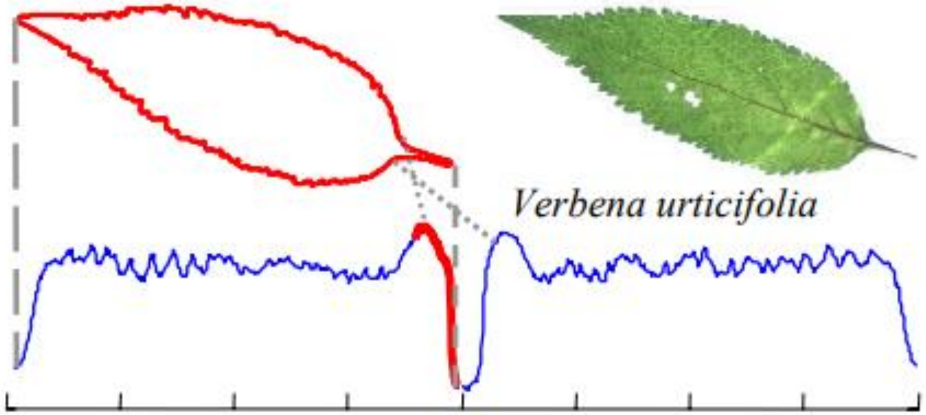
Time series classification – an example

- If we aim to design a classifier to differentiate between the time series of two plant species, what features should we focus on? Given that the **intra-class variability** in color and size significantly exceeds the **inter-class differences**, a viable strategy consists into analyzing the shapes of the leaves.

Time series classification – general intuition

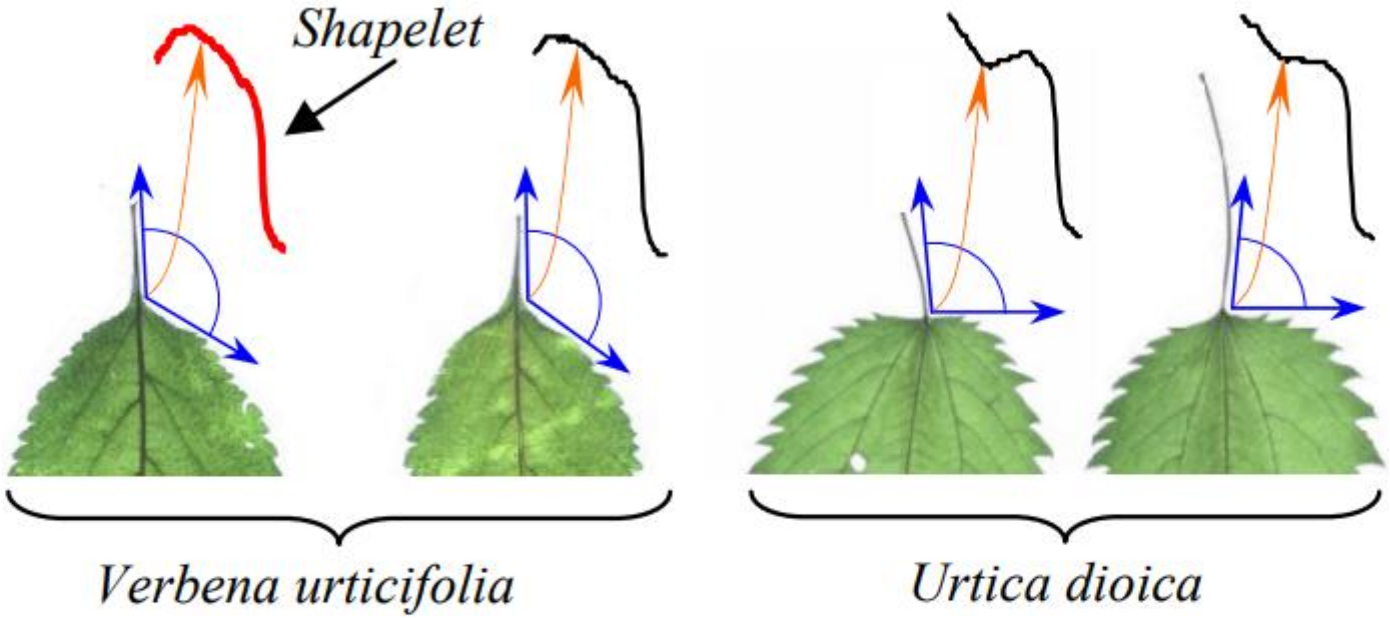
- However, the global shape differences are quite subtle. Moreover, leaves often exhibit distortions or "occlusions" caused by insect damage, which can confound any global shape-based measurements.

- To address this, we propose an alternative approach: converting each leaf into a one-dimensional representation, as demonstrated in the figure below.



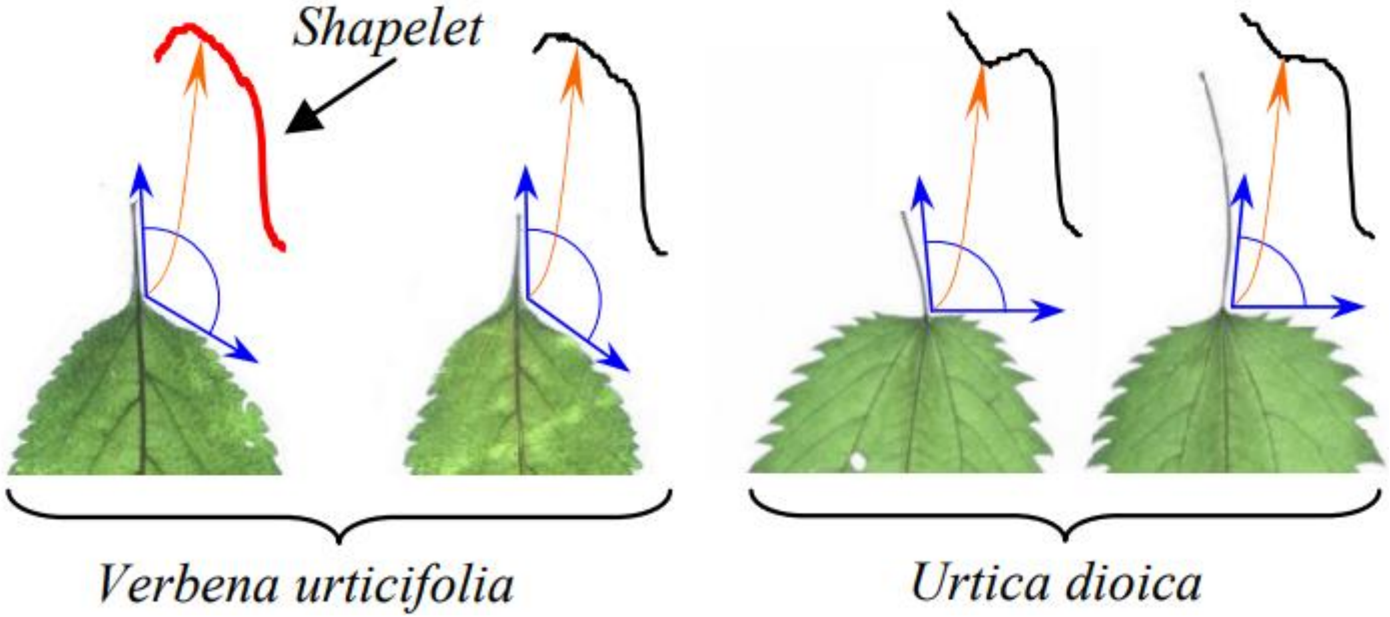
Time series Shapelet

- Instead of comparing all the shapes, we can focus on analyzing a small, highly distinctive subsection of the shapes from the two classes.



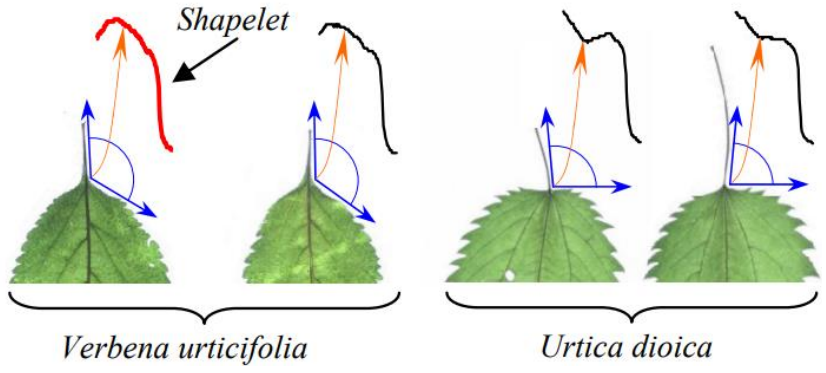
Time series Shapelet

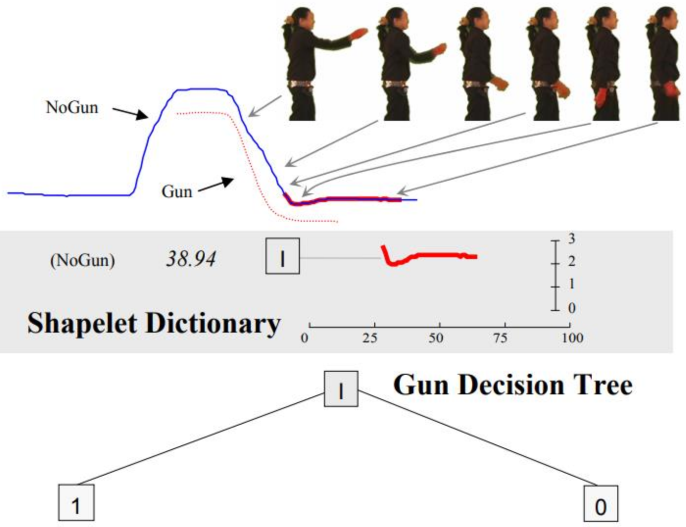
- We call such discriminative subsequence : Time series Shapelet.



Time series Shapelet

- Shapelets were first proposed as time-series segments that maximally predict the target variable. All possible segments were considered as potential candidates. In contrast, the **minimum distances** of a candidate to all training series were used as a predictor feature for ranking the ***information gain*** accuracy of that candidate on the target variable.





Ye, Keogh, Time Series Shapelets: A New Primitive for Data Mining, KDD 2009

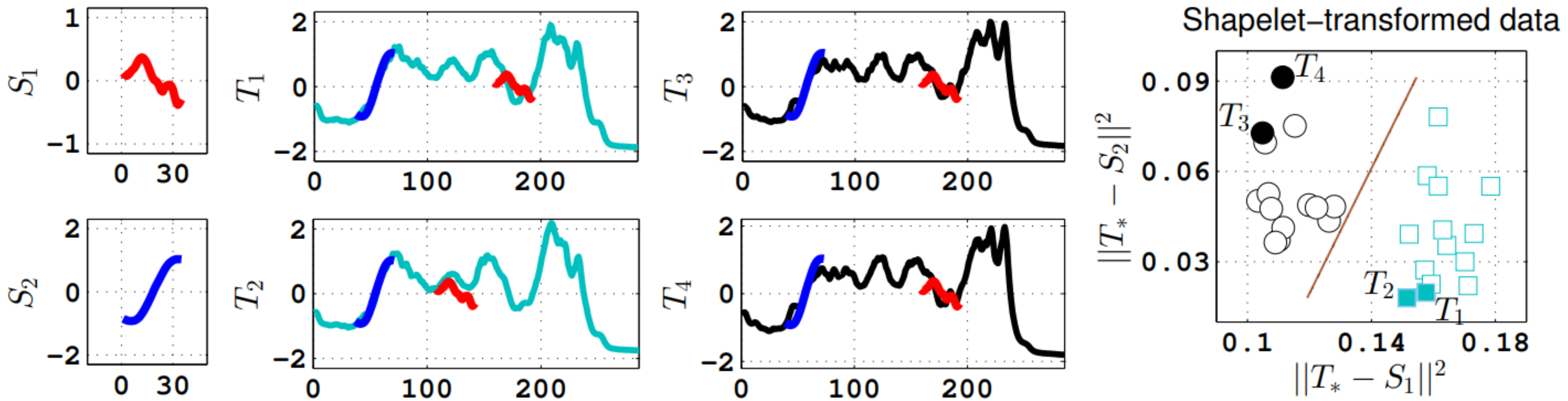
Learning Time series shapelet

1) We can consider a mathematical formulation of the shapelet learning task as an optimization of a classification objective function.

2) Shapelets can be learnt such as their distances to the original series can linearly separate the time series instances by their targets.

Learning Time series shapelet

- Two learned shapelets S1, S2.
- Series' distances to shapelets can optimally project the series into a
- 2-dimensional space, called the shapelet-transformed representation



Notation

- ***Time Series Dataset***: A time-series dataset composed of I training instances, with each series contains Q-many ordered values, is denoted as $T^{I \times Q}$, while the series target is a nominal variable $Y \in \{1, \ldots, C\}^I$ having C categories.

- ***Shapelets***: A shapelet of length L is simply an ordered sequence of values from a data structure perspective. Nevertheless, shapelets semantically represent intelligence on how to discriminate the target variable of a series dataset. The K-most informative shapelets are denoted as $S \in \mathbb{R}^{K \times L}$.

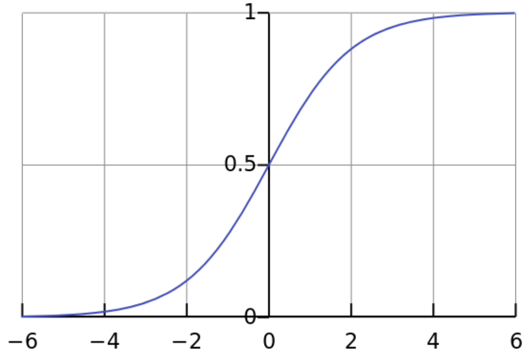
Notation

- ***Shapalet Transformation***: Minimum distances to shapelets can be characterized as a transformation of the time-series data $T \in \mathbb{R}^{I \times Q}$ into a new representation $M \in \mathbb{R}^{I \times K}$. Such a transformation reduces the dimensionality of the original time-series, because typically $K < Q$

- General purpose classifiers (e.g.: SVMs, Bayesian Network, . . .) show high prediction accuracy over the new representation M.

Logistic regression model

- We consider a ***logistic regression*** classification model. It provides an option to interpret predicted binary targets as probabilistic confidences. Nonetheless, it can ensure extending to the multi-class case.



- ***Logistic Sigmoid Function***: $\sigma(Y) = (1 + e^{-Y})^{-1}$.

- It is used for the prediction of target variables via a logistic regression loss.

Learning model

- The minimum distances M become the new predictors in the transformed shapelets space, a ***linear learning model*** can predict approximate target values $\hat{Y} \in \mathbb{R}^{I \times K}$ via the predictors M and linear weights $W \in \mathbb{R}^K$ (plus bias $W_0 \in \mathbb{R}$).

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

Loss Function

- The logistic regression operates by minimizing the ***logistic loss***, between **true targets *Y* and estimated ones $\widehat{\boldsymbol{Y}}$**.

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

Regularized Objective Function

- The logistic loss function together with regularization terms represent the regularized objective function.
- ***The idea is to jointly learn the optimal shapelets S and the optimal linear hyper-plane W that minimize the classification objective $\mathcal{F}$.***

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

Differentiable Soft-Minimum Function

- In order to compute the derivative of the objective function, all the involved functions of the model need to be differentiable.

- We denote the distance between the *j*-th segment of series *i* and the *k*-th shapelet as $D_{i,k,j}$.

$$D_{i,k,j} := \frac{1}{L}\sum_{l=1}^{L}\left(T_{i,j+l-1} - S_{k,l}\right)^2$$

Differentiable Soft-Minimum Function

- A differentiable approximation of the minimum function is the popular *Soft Minimum*.
- A parameter α controls the *precision* of the function and the soft minimum approaches the true minimum for $\alpha \rightarrow -\infty$.

$$M_{i,k} \approx \widehat{M}_{i,k} = \frac{\sum_{j=1}^{J} D_{i,k,j} e^{\alpha D_{i,k,j}}}{\sum_{j=1}^{J} e^{\alpha D_{i,k,j}}}$$

Per-instance objective

- The decomposed objective function $\mathcal{F}_i$ corresponds to a division of the objective function $\mathcal{F}$ into per-instance losses for each time series.

$$\mathcal{F}_i = \mathcal{L}\left(Y_i, \hat{Y}_i\right) + \frac{\lambda_w}{I} \sum_{k=1}^{K} W_k^2$$

Optimization algorithm - Gradient descent

Algorithm 1 Learning Time-Series Shapelets

Require: $T \in \mathbb{R}^{I \times Q}$, Number of Shapelets K, Length of a shapelet L, Regularization λ_W, Learning Rate η, Number of iterations: maxIter

Ensure: Shapelets $S \in \mathbb{R}^{K \times L}$, Classification weights $W \in \mathbb{R}^K$, Bias $W_0 \in \mathbb{R}$

for iteration=$\mathbb{N}_1^{\text{maxIter}}$ **do**
    for $i = 1, \ldots, I$ **do**
        for $k = 1, \ldots, K$ **do**
            $W_k \leftarrow W_k - \eta \frac{\partial \mathcal{F}_i}{\partial W_k}$
            for $L = 1, \ldots, L$ **do**
                $S_{k,l} \leftarrow S_{k,l} - \eta \frac{\partial \mathcal{F}_i}{\partial S_{k,l}}$
            end for
        end for
        $W_0 \leftarrow W_0 - \eta \frac{\partial \mathcal{F}_i}{\partial W_0}$
    end for
end for
return S, W, W_0

References

- **Ye, Keogh, Time Series Shapelets: A New Primitive for Data Mining, KDD 2009 https://dl.acm.org/doi/10.1145/1557019.1557122**
- **Josif Grabocka et al. Learning time-series shapelets Published in Knowledge Discovery and Data 2014, https://www.semanticscholar.org/paper/Learning-time-series-shapelets-Grabocka-Schilling/5900721a7ff9e91782dd4dc33956b09d3adb67f5**