Learning Time-Series Shapelets

Josif Grabocka, Nicolas Schilling, Martin Wistuba, Lars Schmidt-Thieme
{josif, schilling, wistuba, schmidt-thieme}@isml1.uni-hildesheim.de
ISMLL, University of Hildesheim, Germany

KDD 2014 THE 20TH ACM SIGKDD CONFERENCE ON KNOWLEDGE DISCOVERY AND DATA MINING AUGUST 2014 | NEW YORK CITY

Introduction

Shapelets are discriminatory sub-sequences of time series. The distances between shapelets and the closest segments of time series define a new feature representation, known as shapelet-transformation:

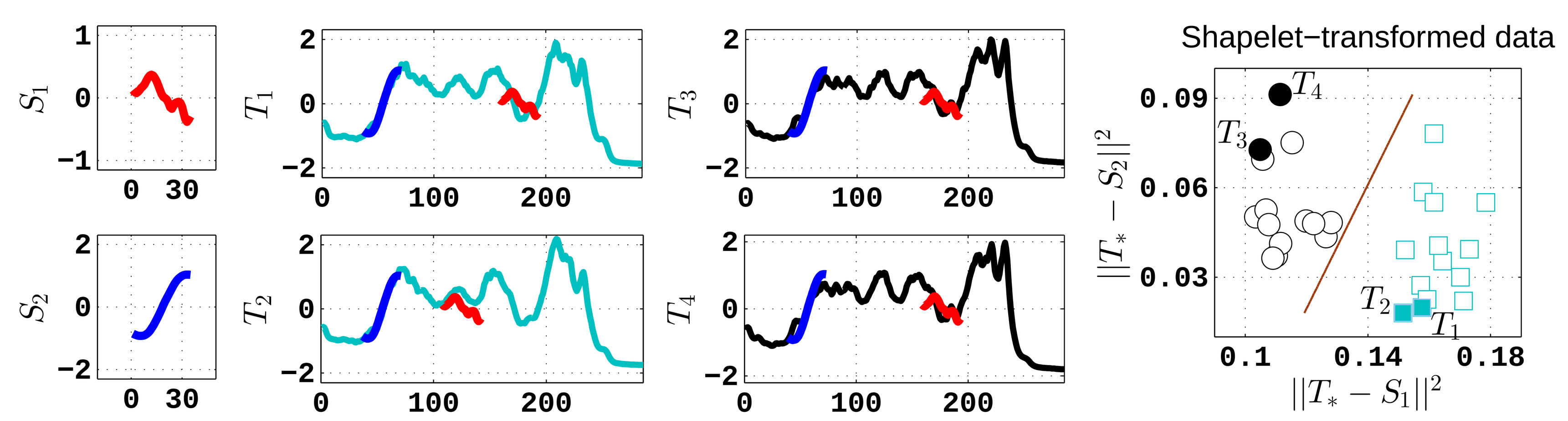

Figure 1: Left: Two shapelets, Middle: Minimum distance matches, Right: 'Shapelet-Transform'ed data

The state-of-the-art finds the discriminative shapelets by exhaustively trying candidate shapelets from segments of training series. In contrast, this paper learns the shapelets by directly optimizing the objective function.

Proposed Method

Assume we are given I-many time-series instances, each having M-many points and denoted as $T \in \mathbb{R}^{I\times M}$. One can partition each series into segments of length L in a sliding window approach and extract $J = M - L$ segments per series. The minimum distances $M \in \mathbb{R}^{I\times K}$ (between K-many shapelets $S \in \mathbb{R}^{K\times L}$ and all the sliding window segments of series T) define the predictors of the derived representation:

$$M_{i,k} = \min_{j=1,\ldots,J} \frac{1}{L}\sum_{l=1}^{L}\left(T_{i,j+l-1} - S_{k,l}\right)^2, \quad \forall i \in \{1,\ldots,I\}, \forall k \in \{1,\ldots,K\} \qquad (1)$$

The predictors $M \in \mathbb{R}^{I\times K}$ can be used to estimate the target $\hat{Y} \in \mathbb{R}^{I}$ of the training instances using a linear model with weights $W \in \mathbb{R}^{K}$ and a bias term $W_0 \in \mathbb{R}$:

$$\hat{Y}_i = W_0 + \sum_{k=1}^{K} M_{i,k} W_k, \quad \forall i \in \{1,\ldots,I\} \qquad (2)$$

The risk of estimating the true target $Y \in \{-1,+1\}^I$ from approximated target $\hat{Y} \in \mathbb{R}^I$ can be measured through a differentiable loss function $\mathcal{L}(Y,\hat{Y}) \in \mathbb{R}^I$, here the logistic loss:

$$\mathcal{L}(Y_i,\hat{Y}_i) = -Y_i \ln \sigma(\hat{Y}_i) - (1-Y_i)\ln\left(1-\sigma(\hat{Y}_i)\right), \quad \forall i \in \{1,\ldots,I\} \qquad (3)$$

The objective function $\mathcal{F} \in \mathbb{R}$ is a regularized loss function, whose output are the shapelets S and classification weights W that achieve the minimum value of $\mathcal{F}$:

$$\underset{S,W}{\operatorname{argmin}}\ \mathcal{F}(S,W) = \underset{S,W}{\operatorname{argmin}} \sum_{i=1}^{I} \mathcal{L}(Y_i,\hat{Y}_i) + \lambda_W ||W||^2 \qquad (4)$$

Since the minimum function M is not differentiable, a derivation of the objective function with respect to shapelets is not feasible. Instead, we use a smooth soft-minimum approximation $\hat{M}$, which can estimate the minimum distance between the k-the shapelet $S_{k,:}$ and all the J-many segments of an arbitrary instance $T_{i,:}$:

$$M_{i,k} \approx \hat{M}_{i,k} = \frac{\sum_{j=1}^{J} D_{i,k,j}\, e^{\alpha D_{i,k,j}}}{\sum_{j'=1}^{J} e^{\alpha D_{i,k,j'}}}, \quad \forall i \in \{1,\ldots,I\}, \forall k \in \{1,\ldots,K\} \qquad (5)$$

$$D_{i,k,j} := \frac{1}{L}\sum_{l=1}^{L}\left(T_{i,j+l-1} - S_{k,l}\right)^2, \quad \forall i \in \{1,\ldots,I\}, \forall k \in \{1,\ldots,K\}, \forall j \in \{1,\ldots,J\} \qquad (6)$$

The smooth approximation of the minimum function, allows only the minimum segment to contribute for $\alpha \to -\infty$.

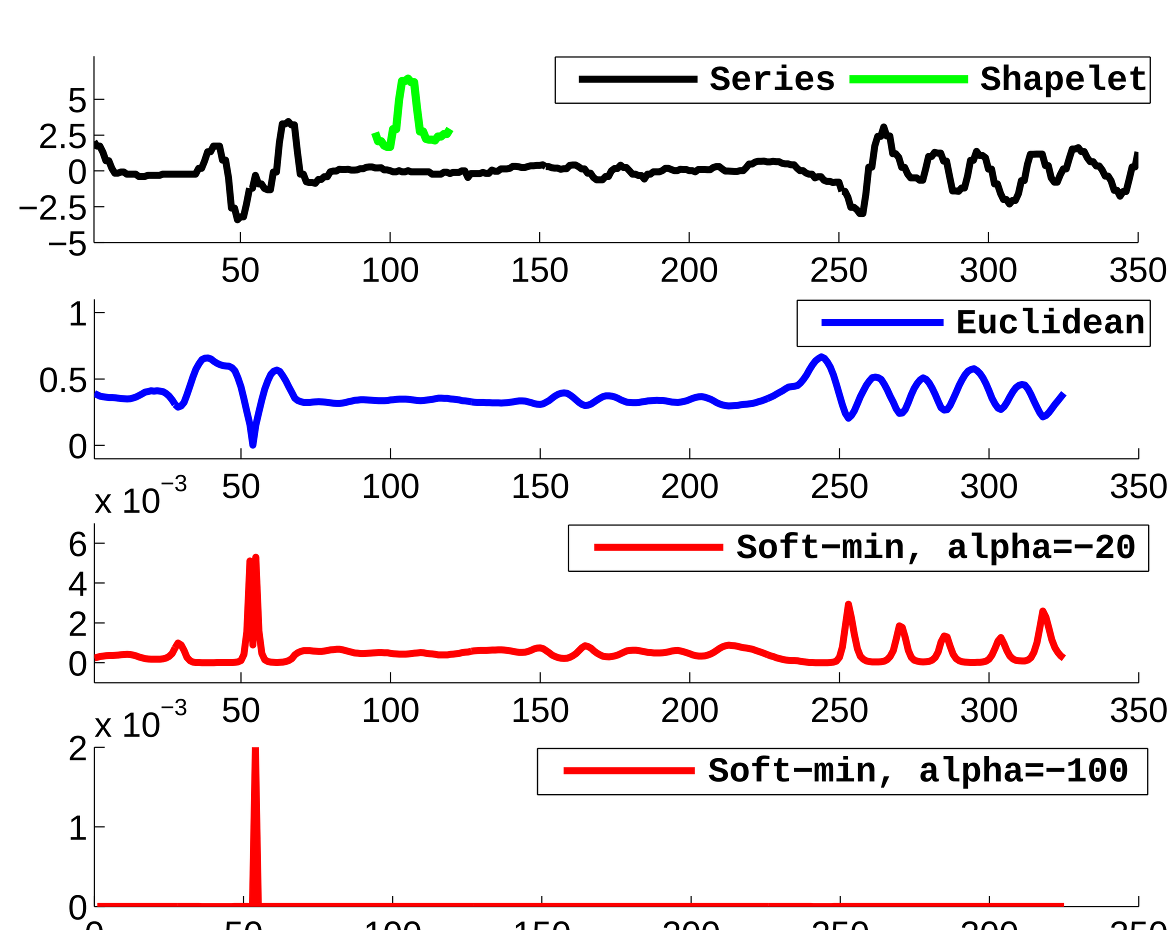

Figure 2: Illustration of the soft minimum between a shapelet (green) and all the segments of a series (black) from the FaceFour dataset

In addition, the objective function $\mathcal{F}$ can be decomposed into per-instance objectives $\mathcal{F}_i$, in order to enable the usage of stochastic gradient descent optimization:

$$\mathcal{F}_i = \mathcal{L}(Y_i,\hat{Y}_i) + \frac{\lambda_W}{I}\sum_{k=1}^{K} W_k^{\,2}, \quad \forall i \in \{1,\ldots,I\} \qquad (7)$$

Learning Algorithm

The partial derivative of the per-instance objective function $\mathcal{F}_i$ with respect to the l-th point of the k-th shapelet $S_{k,l}$ is computed using the chain rule of derivation:

$$\frac{\partial \mathcal{F}_i}{\partial S_{k,l}} = \frac{\partial \mathcal{L}(Y_i,\hat{Y}_i)}{\partial \hat{Y}_i}\frac{\partial \hat{Y}_i}{\partial \hat{M}_{i,k}}\sum_{j=1}^{J}\frac{\partial \hat{M}_{i,k}}{\partial D_{i,k,j}}\frac{\partial D_{i,k,j}}{\partial S_{k,l}} \qquad (8)$$

All the components of the partial derivative are computable as follows:

$$\frac{\partial \mathcal{L}(Y_i,\hat{Y}_i)}{\partial \hat{Y}_i} = -\left(Y_i - \sigma\left(\hat{Y}_i\right)\right), \quad \frac{\partial \hat{Y}_i}{\partial \hat{M}_{i,k}} = W_k \qquad (9)$$

$$\frac{\partial \hat{M}_{i,k}}{\partial D_{i,k,j}} = \frac{e^{\alpha D_{i,k,j}}\left(1+\alpha\left(D_{i,k,j}-\hat{M}_{i,k}\right)\right)}{\sum_{j'=1}^{J} e^{\alpha D_{i,k,j'}}}, \quad \frac{\partial D_{i,k,j}}{\partial S_{k,l}} = \frac{2}{L}\left(S_{k,l} - T_{i,j+l-1}\right) \qquad (10)$$

In addition, the partial derivative of the per-instance objective with respect to each k-th cell of the weights vector W and the bias W_0 are derived as follows:

$$\frac{\partial \mathcal{F}_i}{\partial W_k} = -\left(Y_i - \sigma\left(\hat{Y}_i\right)\right)\hat{M}_{i,k} + \frac{2\lambda_W}{I}W_k, \quad \frac{\partial \mathcal{F}_i}{\partial W_0} = -\left(Y_i - \sigma\left(\hat{Y}_i\right)\right) \qquad (11)$$

Ultimately, a stochastic gradient descent learning algorithm can be conducted to learn the shapelets S and the classification weights W:

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

The learning algorithm updates the shapelets, such that the minimum distances M and the weights W, W_0 ensure a minimal classification loss value:

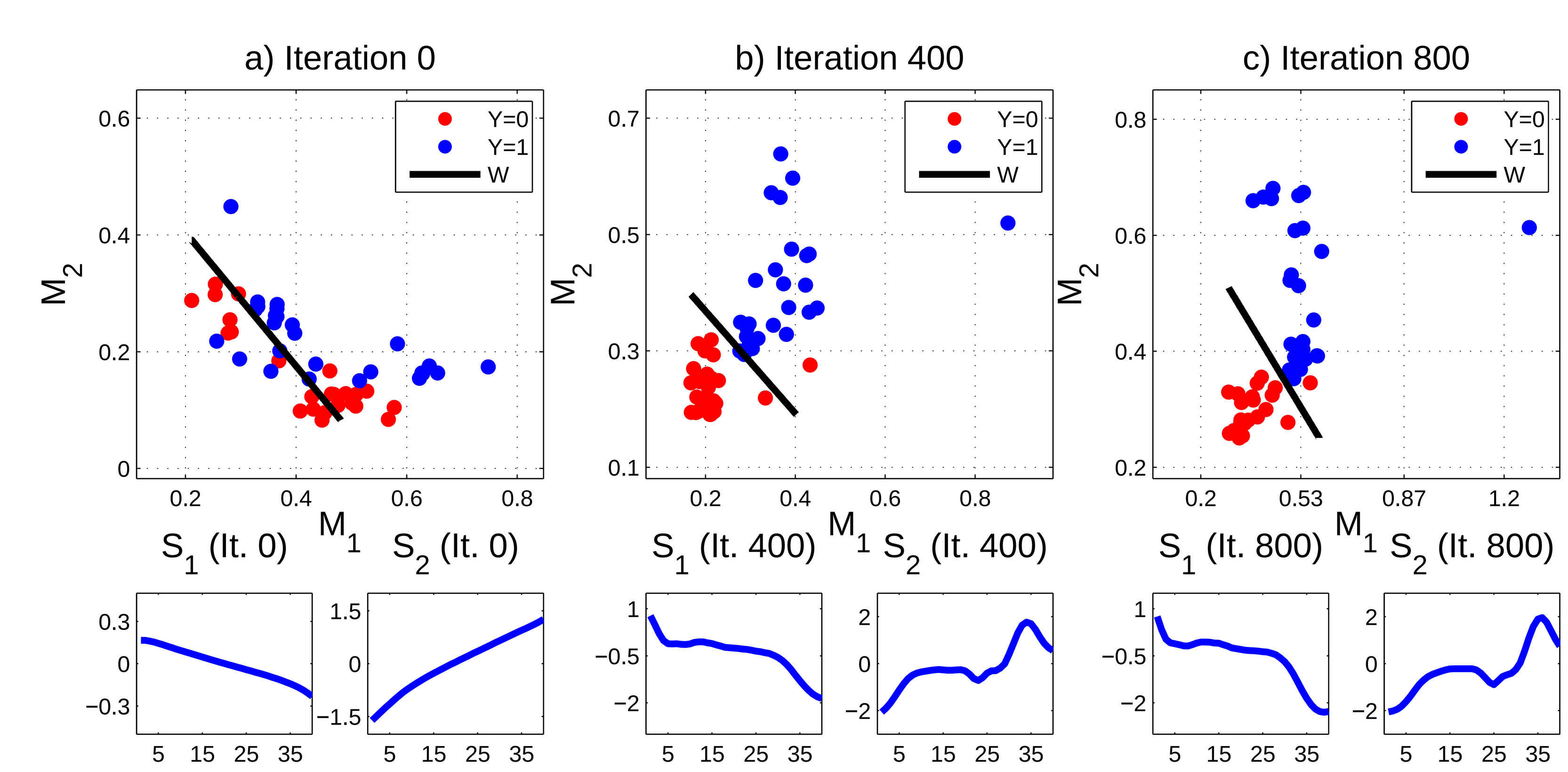

Figure 3: Learning Two Shapelets on the Gun-Point Dataset, ($L = 40, \eta = 0.01, \lambda_W = 0.01, \alpha = -100$)

Empirical Results and Conclusions

- Baselines: **Quality Criteria**: Information gain (IG), Kruskall-Wallis (KW), F-Stats (FST), Mood's Median Criterion (MM); **Using shapelet-transformed data**: Nearest Neighbors (1NN), Naive Bayes (NB), C4.5 tree (C4.5), Bayesian Networks (BN), Random Forest (RAF), Rotation Forest (ROF), Support Vector Machines (SVM); **Other Related**: Fast Shapelets (FSH), **Dynamic Time Warping** (DTW).
- Datasets: 28 time-series datasets of the UCR and UEA collections collected from diverse domains

	IG	KW	FST	MM	DTW	C4.5	NN	NB	BN	RAF	ROF	SVM	FSH	LTS
Absolute Wins	0.00	0.00	1.20	0.25	0.00	0.00	1.53	0.00	1.58	0.20	0.00	4.70	1.25	17.28
LTS Wins	28	27	26	26	28	28	23	27	23	26	24	20	26	-
LTS Draws	0	0	1	1	0	0	2	0	2	1	1	2	1	-
LTS Losses	0	1	1	1	0	0	3	1	3	1	3	6	1	-
Rank Mean	9.768	9.982	9.107	9.500	9.804	10.036	6.196	7.714	5.518	6.321	5.357	4.554	9.196	1.946
Rank C.I.	1.016	1.259	1.318	1.273	1.867	0.781	1.195	1.091	1.150	0.743	0.898	1.180	1.519	0.536
Rank St.Dev.	2.743	3.398	3.559	3.436	5.040	2.108	3.228	2.944	3.104	2.005	2.423	3.186	4.102	1.448
Wil. p-values	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.003	0.000	-

Table 1: Comparative Figures of Classification Accuracies over 28 Time-series Datasets

Results indicate that shapelets which are learned to optimize the objective function directly produce significantly better classification accuracies compared to shapelets learned from exhaustive aproaches.