Learning Time-Series Shapelets

Josif Grabocka, Nicolas Schilling, Martin Wistuba and Lars Schmidt-Thieme

Information Systems and Machine Learning Lab (ISMLL)
University of Hildesheim, Germany

SIGKDD 2014, 26/08/2014
Learning Time-Series Shapelets

What are Time-Series Shapelets? (I)

Definition:
- **Patterns** whose minimum distances to time-series yield discriminative predictors [Ye and Keogh(2009), Lines et al.(2012)Lines, Davis, Hills, and Bagnall]

Problem:
1. **Learn** $K$ discriminative shapelets of length $L$ (denoted as $S \in \mathbb{R}^{K \times L}$).
2. **From** a dataset that has $I$ time-series instances of length $M$ (denoted as $T \in \mathbb{R}^{I \times M}$), where each series is divided into $J = M - L$ sliding window segments.
What are Time-Series Shapelets? (II)

The minimum distances $M \in \mathbb{R}^{I \times K}$ between shapelets and time-series:

$$M_{i,k} = \min_{j=1, \ldots, J} \frac{1}{L} \sum_{l=1}^{L} (T_{i,j+l-1} - S_{k,l})^2$$

(1)

... yield discriminative predictors:

Figure 1: Left: Two shapelets, Middle: Closest Segments, Right: 'Shapelet-Transformed data'

Grabocka et al., ISMLL, University of Hildesheim, Germany
Related Shapelets Work

**All** the possible segments (*exhaustively*) of all time-series candidates are potential shapelet candidates:

- Compute the prediction accuracy of minimum distances of each candidate and then **rank** the top-K by prediction accuracy (feature ranking)
- Build a decision tree from the top-K features [Ye and Keogh(2009), Lines et al.(2012)Lines, Davis, Hills, and Bagnall]
- Alternatively, use the new K-dimensional feature representation and use **standard classifiers** [Hills et al.(2013)Hills, Lines, Baranauskas, Mapp, and Bagnall].

**Exhaustive** search is **expensive**, speed-ups were proposed [Mueen et al.(2011)Mueen, Keogh, and Young, Rakthanmanon and Keogh(2013)].
Proposed Method (I)

A linear model of predictors $M \in \mathbb{R}^{I \times K}$ and weights $W \in \mathbb{R}^K$, $W_0 \in \mathbb{R}$ can be used to estimate the target $\hat{Y} \in \mathbb{R}^I$:

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

(2)

The risk of estimating the true target $Y \in \{-1, +1\}^I$ from approximated target $\hat{Y} \in \mathbb{R}^I$ is the logistic loss $\mathcal{L}(Y, \hat{Y}) \in \mathbb{R}^I$:

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

(3)
Proposed Method (II)

The objective function $\mathcal{F} \in \mathbb{R}$ is a regularized loss function:

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

The objective function $\mathcal{F}$ can be decomposed into per-instance objectives $\mathcal{F}_i$:

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

Mission of This Paper: Learn $S, W$ that minimize $\mathcal{F}$. 
Differentiable Minimum Function (I)

Approximate the true minimum $M$ with the soft-minimum version $\hat{M}$:

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

$$\alpha \in (-\infty, 0] \quad \forall i \in \{1, \ldots, I\}, \forall k \in \{1, \ldots, K\} \quad (6)$$

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

$$\forall i \in \{1, \ldots, I\}, \forall k \in \{1, \ldots, K\}, \forall j \in \{1, \ldots, J\} \quad (7)$$
Differentiable Minimum Function (II)

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

![Graph illustrating the soft minimum between a shapelet (green) and all the segments of a series (black) from the FaceFour dataset](image)

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

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

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

(8)

$$ \frac{\partial F_i}{\partial W_k} = \frac{\partial L(Y_i, \hat{Y}_i)}{\partial \hat{Y}_i} \frac{\partial \hat{Y}_i}{\partial \hat{W}_k} + \frac{\partial \text{Reg}(W)}{\partial W_k}, \quad \frac{\partial F_i}{\partial W_0} = \frac{\partial L(Y_i, \hat{Y}_i)}{\partial \hat{Y}_i} $$

(9)

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

$$ \frac{\partial L(Y_i, \hat{Y}_i)}{\partial \hat{Y}_i} = -(Y_i - \sigma(\hat{Y}_i)) $$, \quad \frac{\partial \hat{Y}_i}{\partial \hat{M}_{i,k}} = W_k, \quad \frac{\partial \hat{Y}_i}{\partial \hat{W}_k} = M_{i,k}, \quad \frac{\partial \text{Reg}(W)}{\partial W_k} = \frac{2\lambda_W}{l} W_k $$

(10)

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

(11)
Learning Algorithm (II)

Require: \( T \in \mathbb{R}^{I \times Q} \), Number of Shapelets \( K \), Length of a shapelet \( L \), Regularization \( \lambda_W \), Learning Rate \( \eta \), Number of iterations: \( \text{maxIter} \)

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

1: for \( \text{iteration}=1,\ldots,\text{maxIter} \) do
2: for \( i = 1,\ldots,I \) do
3: \( W_0 \leftarrow W_0 - \eta \frac{\partial F_i}{\partial W_0} \)
4: for \( k = 1,\ldots,K \) do
5: \( W_k \leftarrow W_k - \eta \frac{\partial F_i}{\partial W_k} \)
6: for \( L = 1,\ldots,L \) do
7: \( S_{k,l} \leftarrow S_{k,l} - \eta \frac{\partial F_i}{\partial S_{k,l}} \)
8: return \( S, W, W_0 \)
Illustration of Learning

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

1. Discover hidden/latent shapelets
2. Interactions of Variables

Figure 4: Interactions among Shapelets Enable Individually Unsuccessful Shapelets (left plots) to Excel in Cooperation (right plot)
Experimental Setup

Our method, denoted LS, is compared against 13 baselines using 28 time-series dataset:

- **Baselines:** Quality Criteria: Information gain (IG), Kruskall-Wallis (KW), F-Stats (FT), Mood’s Median Criterion (MM); Using shapelet-transformed data: Nearest Neighbors (1NN), Naive Bayes (NB), C4.5 tree (C4), Bayesian Networks (BN), Random Forest (RA), Rotation Forest (RO), Support Vector Machines (SV); Other Related: Fast Shapelets (FS), Dynamic Time Warping (DT).

- **Datasets:** 28 time-series datasets from the UCR and UEA collections from diverse domains; Provided train/test splits

- **Hyper-parameters:** are found using grid-search, by testing over a validation split from the training data
## Results

<table>
<thead>
<tr>
<th></th>
<th>IG</th>
<th>KW</th>
<th>FT</th>
<th>MM</th>
<th>DT</th>
<th>C4</th>
<th>NN</th>
<th>NB</th>
<th>BN</th>
<th>RF</th>
<th>RO</th>
<th>SV</th>
<th>FS</th>
<th>LS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Wins</td>
<td>0.0</td>
<td>0.0</td>
<td>1.2</td>
<td>0.3</td>
<td>0.0</td>
<td>0.0</td>
<td>1.5</td>
<td>0.0</td>
<td>1.6</td>
<td>0.2</td>
<td>0.0</td>
<td>4.7</td>
<td>1.3</td>
<td>17.3</td>
</tr>
<tr>
<td>LS Wins</td>
<td>28</td>
<td>27</td>
<td>26</td>
<td>26</td>
<td>28</td>
<td>28</td>
<td>23</td>
<td>27</td>
<td>23</td>
<td>26</td>
<td>24</td>
<td>20</td>
<td>26</td>
<td>-</td>
</tr>
<tr>
<td>LS Draws</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>LS Losses</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>6</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>Rank Mean</td>
<td>9.8</td>
<td>9.9</td>
<td>9.1</td>
<td>9.5</td>
<td>9.8</td>
<td>10.0</td>
<td>6.2</td>
<td>7.7</td>
<td>5.5</td>
<td>6.3</td>
<td>5.4</td>
<td>4.6</td>
<td>9.2</td>
<td>1.9</td>
</tr>
<tr>
<td>Rank C.I.</td>
<td>1.0</td>
<td>1.3</td>
<td>1.3</td>
<td>1.3</td>
<td>1.9</td>
<td>0.8</td>
<td>1.2</td>
<td>1.1</td>
<td>1.2</td>
<td>0.7</td>
<td>0.9</td>
<td>1.2</td>
<td>1.5</td>
<td>0.5</td>
</tr>
<tr>
<td>Rank S.D.</td>
<td>2.7</td>
<td>3.4</td>
<td>3.6</td>
<td>3.4</td>
<td>5.0</td>
<td>2.1</td>
<td>3.2</td>
<td>2.9</td>
<td>3.1</td>
<td>2.0</td>
<td>2.4</td>
<td>3.2</td>
<td>4.1</td>
<td>1.4</td>
</tr>
<tr>
<td>W.t. p-val.</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>-</td>
</tr>
</tbody>
</table>

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

1. Learning shapelets to directly optimize the classification objective improves classification accuracy

2. Supervised Shapelet Learning more accurate than Exhaustive Shapelet Discovery
   ▶ Shapelets are not restricted to series segments
   ▶ Considers interactions among minimum distances features

3. Results against 13 baselines using 28 datasets validate the claim
References


Back-up Slide: Dependence on Initialization

Figure 5: Sensitivity of Shapelet Initialization, Gun-Point dataset, Parameters:
$L = 30, \eta = 0.01, \lambda_W = 0.01, \text{Iterations} = 3000, \alpha = -100$

- We used K-Means centroids as initial shapelets.