Cost Sensitive Time-series classification

Shoumik Roychoudhury, Mohamed Ghalwash, Zoran Obradovic

Center for Data Analytics and Biomedical Informatics, Temple University

shoumik.rc, mohamed.ghalwash, zoran.obradovic (@temple.edu)

Imbalanced Time-series Classification

Introduction: One of the key sources of performance degradation in the field of time-series classification is the class imbalance problem. In real-world datasets, the minority class (Positive class) is outnumbered by abundant majority (negative) class instances.

Objective: Develop a cost-sensitive time-series classification framework.

Challenge 1: Minimum classification error criterion (e.g 0-1 loss function) based classification models generate biased models towards the majority class causing higher misclassification error for minority class examples (important class).

Solution: Use differentially weighted loss function having variable misclassification cost for false positive and false negative errors.

Challenge 2: Predetermination of misclassification cost values from domain experts.

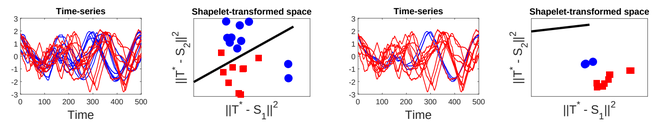
Solution: Estimating misclassification cost from data.

Challenge 3: Interpretable Model.

Solution: Learning interpretable temporal patterns (shapelets) for time-series classification.

Background: Time-series Classification

- Learning Time-series Shapelets (LTS) [Grabocka et al. (KDD 2014)] is state-of-the-art model for learning shapelets for time-series classification.
- Shapelets are short time-series sub-sequences (S_1, S_2)
- T^* is the time-series dataset.



- In LTS, a cost-insensitive 0-1 logistic loss function is minimized in order to learn generalized shapelets.

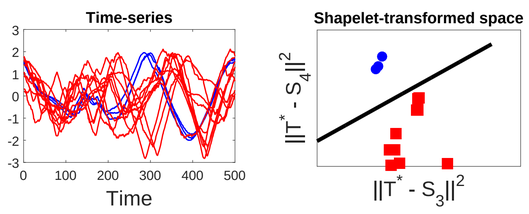
$$\mathcal{L}(Y, \hat{Y}) = -Yln\sigma(\hat{Y}) - (1-Y)ln(1-\sigma(\hat{Y})) \quad (1)$$

- Linear Model jointly learning weights of classification hyperplane and shapelets:

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

- The minimum Euclidean distances $M_{i,k}$ of the learned shapelets S_1, S_2 to the time-series T^* is used as features to linearly separate the examples in the shapelet-transformed space.

Proposed method: Cost sensitive Time-series classification (CS-LTS)



- Cost sensitive extension of learning function:

$$Z_i = \frac{1}{C_{FN} + C_{FP}} ln \frac{\sigma(\hat{Y}) C_{FN}}{1 - \sigma(\hat{Y}) C_{FP}} = \frac{1}{C_{FN} + C_{FP}} (\hat{Y} + ln \frac{C_{FN}}{C_{FP}}) \quad (3)$$

- C_{FP}, C_{FN}: Misclassification cost for false positive error and false negative error. They denote the loss incurred when a wrong prediction occurs.
- cost-sensitive logistic loss function:

$$\mathcal{L}(Y, Z) = -Yln\sigma(C_{FN} Z) - (1-Y)ln(1-\sigma(C_{FP} Z)) \quad (4)$$

- Regularized Objective function:

$$\underset{S,W,C}{\operatorname{argmin}} \mathcal{F}(S, W, C) = \underset{S,W,C}{\operatorname{argmin}} \sum_{i=1}^{I} \mathcal{L}(Y_i, Z_i) + \lambda_W \|W\|^2 \quad (5)$$

where $C \in \{C_{FN}, C_{FP}\}$

- Constrained optimization problem:

$$\underset{S,W,C}{\operatorname{argmin}} \mathcal{F}(S, W, C) \quad \text{subject to } C_{FN} > 0,\ C_{FP} > 0,\ C_{FN} > \theta C_{FP} \quad (6)$$

- Misclassification costs should always be positive.
- Cost of false negative is at least θ times greater than cost of false positive. This conditions ensures the loss function to be penalized more in the event of an error in the positive class compared to an error in the negative class.

Learning Algorithm: CS-LTS

- Stochastic Gradient descent algorithm used to solve the optimization problem.
- Gradients are computed as partial derivatives of the per-instance function:

$$\mathcal{F}_i = \mathcal{L}(Y_i, Z_i) + \frac{\lambda_W}{I} \sum_{k=1}^{K} W_k^2 \quad (7)$$

- Following equation shows the point gradient of objective function for the i^{th} time-series with respect to shapelet S_k

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

- Partial derivatives of each component are as follows:

$$\frac{\partial \mathcal{L}(Y_i, Z_i)}{\partial Z_i} = (1-Y_i)\sigma(C_{FP} Z_i) C_{FP} - Y_i (1 - \sigma(C_{FN} Z_i)) C_{FN} \quad (9)$$

$$\frac{\partial \hat{Y}_i}{\partial \hat{M}_{i,k}} = W_k \quad (10)$$

$$\frac{\partial \hat{M}_{i,k}}{\partial D_{i,k,j}} = \frac{\exp(\alpha D_{i,k,j} (1 + \alpha (D_{i,k,j} - \hat{M}_{i,k}))}{\sum_{\bar{j}=1}^{J} \exp(\alpha D_{i,k,\bar{j}})} \quad (11)$$

$$\frac{\partial D_{i,k,j}}{\partial S_{k,l}} = \frac{2}{L} (S_{k,l} - T_{i,j+l-1}) \quad (12)$$

- The gradients with respect to weights of the hyperplane are:

$$\frac{\partial \mathcal{F}_i}{\partial W_k} = \frac{\partial \mathcal{L}(Y_i, Z_i)}{\partial Z_i} \frac{\partial Z_i}{\partial \hat{Y}_i} \hat{M}_{i,k} + \frac{2\lambda_W}{I} W_k \quad (13)$$

$$\frac{\partial \mathcal{F}_i}{\partial W_0} = \frac{\partial \mathcal{L}(Y_i, Z_i)}{\partial Z_i} \frac{\partial Z_i}{\partial \hat{Y}_i} \quad (14)$$

- Estimation of the misclassification cost values is a constrained optimization problem. $C_{FN} > 0$, $C_{FP} > 0$ and $C_{FN} > \theta C_{FP}$, where $\theta \in \mathbb{Z}$
- The constrained optimization is converted into an unconstrained optimization.
- False negative misclassification cost is written in terms of false positive misclassification cost.

$$C_{FN} = \theta C_{FP} + \mathcal{D} \quad (15)$$

- Revised objective function:

$$\underset{S,W,C_{FP},\mathcal{D}}{\operatorname{argmin}} \mathcal{F}(S, W, C_{FP}, \mathcal{D}) \quad \text{subject to } C_{FP} > 0 \quad (16)$$

- Gradient with respect to false positive error cannot be optimized using Stochastic gradient descent.
- Gradient is computed with respect to logarithm of false positive cost which allows us to convert the constrained formulation to an unconstrained problem.

$$\frac{\partial \mathcal{F}_i}{\partial \log c_{FP}} = c_{FP} \frac{\partial \mathcal{L}(Y_i, Z_i)}{\partial c_{FP}}, \frac{\partial \mathcal{L}(Y_i, Z_i)}{\partial c_{FP}} = \frac{\partial \mathcal{L}(Y_i, Z_i)}{\partial Z_i} \frac{\partial Z_i}{\partial c_{FP}} \quad (17)$$

$$\frac{\partial \mathcal{L}(Y_i, Z_i)}{\partial c_{FP}} = \frac{\partial \mathcal{L}(Y_i, Z_i)}{\partial Z_i} \frac{\partial Z_i}{\partial c_{FP}} \quad (18)$$

$$\frac{\partial \mathcal{L}(Y_i, Z_i)}{\partial \mathcal{D}} = \frac{\partial \mathcal{L}(Y_i, Z_i)}{\partial Z_i} \frac{\partial Z_i}{\partial \mathcal{D}} \quad (19)$$

- The algorithm for the whole procedure is shown:

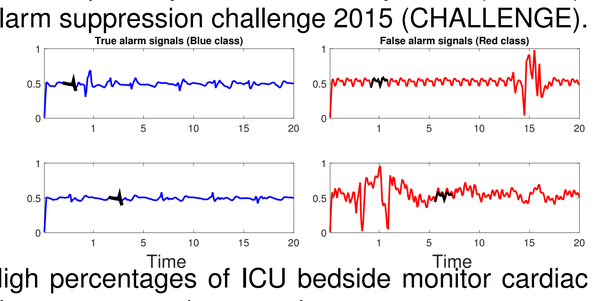
Algorithm 1 Cost-sensitive learning time-series shapelets

procedure CS-LTS
Input: $T \in \mathcal{R}^{I \times Q}$, Number of shapelets K, length of a shapelet L, Regularization parameter λ_W, Learning rate η, maxIter
Initialize: Shapelets $S \in \mathbb{R}^{K \times L}$, classification hyperplane weights $W \in \mathbb{R}^K$, Bias $W_0 \in \mathbb{R}$, Misclassification cost $C_{FP} \in \mathbb{R}$, $\theta \in \mathbb{Z}$, $\mathcal{D} \in \mathbb{R}$
  for iterations = $\mathbb{N}_1^{maxIter}$ **do**
    for $i = 1, ..., I$ **do**
      for $k = 1, ..., K$ **do**
        $W_k^{new} \leftarrow W_k^{old} - \eta \frac{\partial \mathcal{F}_i}{\partial W_k}$
        for $l = 1, ..., L$ **do**
          $S_{k,l}^{new} \leftarrow S_{k,l}^{old} - \eta \frac{\partial \mathcal{F}_i}{\partial S_{k,l}}$
        $W_0^{new} \leftarrow W_0^{old} - \eta \frac{\partial \mathcal{F}_i}{\partial W_0}$
        $\log C_{FP}^{new} \leftarrow \log C_{FP}^{old} - \eta \frac{\partial \mathcal{F}_i}{\partial \log C_{FP}}$
        $\mathcal{D}^{new} \leftarrow \mathcal{D}^{old} - \eta \frac{\partial \mathcal{F}_i}{\partial \mathcal{D}}$
  Return S, W, W_0, C_{FP}

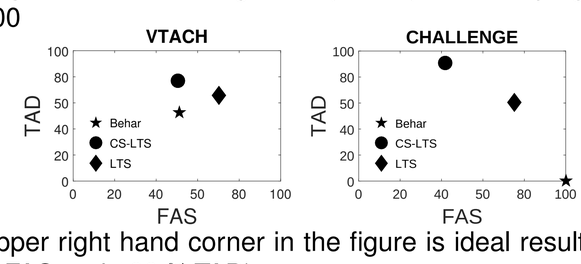
- **Contribution:** A constrained optimization problem which jointly learns shapelets (highly interpretable patterns), hyperplane weights, and most importantly misclassification costs, while other cost-sensitive approaches mainly consider misclassification costs are given a priori.

Experimental Results and Conclusion

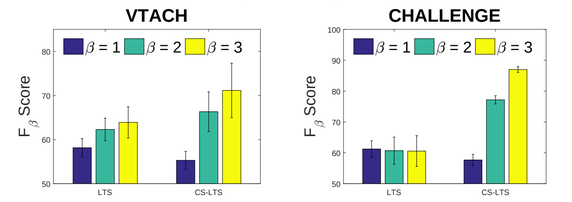
- **Suppression of false Cardiac Arrhythmia alarms in ICU patients**
- Two critical arrhythmia alarm datasets from MIMIC II version 3 repository. Ventricular tachycardia (VTACH) False alarm suppression challenge 2015 (CHALLENGE).



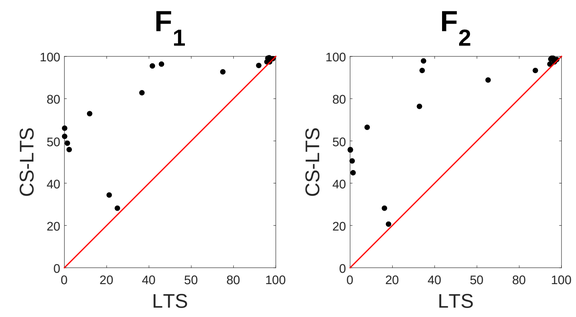
- High percentages of ICU bedside monitor cardiac false alarms compared to true alarms.
- **Objective**: Achieve high FAS (X-axis) while keeping near 100



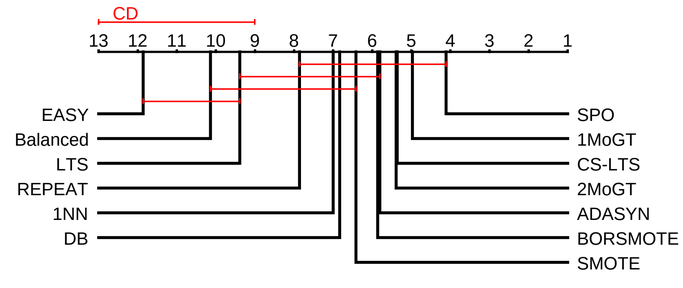
- Upper right hand corner in the figure is ideal result.(100 % FAS and 100 % TAD).
- Proposed method CS-LTS (**Circle**) outperform all baseline methods in terms of TAD in both the datasets.
- Baseline methods are better in terms of FAS however they make lot of false negative errors.
- Agenda: True positives, false negatives and false positives should not have equal weights.



- Higher true positive means lesser missed true alarms. False negative errors are more costlier than false positives. False negative might result in patient death (missed true alarm).
- False negative errors are penalized more using F_β. For example, β = 2, 3 penalizes false negative error more and awards true positive more than β = 1 which represents harmonic mean.
- Conclusion: Increased true alarm detection rates with sufficiently high false alarms suppression rates.
- **Highly imbalanced time-series datasets**
- Agenda: Advantage of cost-sensitive learning over cost-insensitive learning.
- 18 highly imbalanced datasets generated from 5 multi-class datasets from UCR archives.



- Conclusion: CS-LTS outperforms or comparable in terms of F_1 and F_2 on 18 highly imbalanced datasets.
- Proposed CS-LTS compared with 12 baseline methods.
- CS-LTS method attains the highest number of absolute wins (5.86 wins).
- Critical difference diagram showing average rank of CS-LTS against all baseline methods on 18 imbalanced datasets.



- CS-LTS is comparable or better than all baseline methods.

Acknowledgments

This research was supported in part by DARPA grant, the National Science Foundation and Temple University Data Science Targeted Funding Program.