九州大学学術情報リポジトリ
Kyushu University Institutional Repository

Time Series Classification Based on Random Shapelets

Suehiro, Daiki
Faculty of Arts and Science, Kyushu University

Kuwahara, Kengo
Department of Informatics, Kyushu University

Hatano, Kohei
Research and Development Division, Kyushu University Library

Takimoto, Eiji
Department of Informatics, Kyushu University

https://hdl.handle.net/2324/1812930

出版情報：NIPS Time Series Workshop. 2016, 2016-12-09
バージョン：
権利関係：

KYUSHU UNIVERSITY

Time Series Classification Based on Random Shapelets

Daiki Suehiro SUEHIRO93@GMAIL.COM
Faculty of Arts and Science
Kyushu University
Kengo Kuwahara KENGO.KUWAHARA@INF.KYUSHU-U.AC.JP
Department of Informatics
Kyushu University
Kohei Hatano HATANO@INF.KYUSHU-U.AC.JP
Research and Development Division, Library
Kyushu University
Eiji Takimoto EIJI@INF.KYUSHU-U.AC.JP
Department of Informatics
Kyushu University

Editor:

Abstract

Time-Series Shapelets are shown to be useful features to learn accurate classifiers for various time series. There are many algorithms for searching or optimizing shapelets, however, they use limited classes of features based on subsequences in the training data. In this paper, we consider an infinitely large set of sequences as a class of shapelets and discuss the generalization bound of the shapelets-based hypothesis class. In practice, instead of using infinitely many sequences, we propose a heuristic approximate algorithm using random sequences that allows us to obtain good classification rules and shapelets. In preliminary experiments over real data sets, we obtained accurate classifiers which are comparable to the state-of-the-art methods.

Keywords: time series classification, shapelets, sparse modeling

1. Introduction

Time-series classification using time-series shapelets has received much more attention lately, where time series shapelets (or, simply shapelets) are features of time series data based on patterns (we will describe a precise definition later). Shapelets-based time-series analyses achieve not only highly accurate classification, but also they are easy to interpret for domain experts. The concept of time-series shapelets was first introduced by Ye and Keogh (2009). The algorithm finds shapelets by using the information gains of potential candidates associated with all the subsequences of the given time series and constructs a decision tree.

Hills et al. (2014) proposed the shapelet transform, which maps a time series to a feature vector $\mathbf{x} \in \mathbb{R}^K$ using K shapelets. Each attribute x_k of the feature vector corresponds to the minimum distance between the shapelet and any subsequences of the time series. The feature map allows us to employ more effective learning algorithms than decision trees, such as SVMs or random forests. Afterward, many algorithms have been proposed to search the shapelets efficiently keeping high prediction accuracy in practice (Karlsson et al., 2016; Grabocka et al., 2015; Renard et al., 2015; Wistuba et al., 2015; Keogh and Rakthanmanon, 2013). The above algorithms are based on the idea that discriminative subsequences as shapelets are contained in training data. But why do they use subsequences? The overfitting can be a great concern since the feature spaces of their methods are restricted by training sample. Basically, shapelets should be selected from any of sequences.

Grabocka et al. (2014) proposed Learning Time-Series Shapelets (LTS) algorithm which took a different approach from subsequence-based algorithms. LTS approximately solves an optimization problem of learning the best shapelets directly without searching subsequences in a brute-force way. In contrast to the above subsequence-based methods, LTS finds nearly optimal shapelets and achieves higher prediction accuracy than the other existing methods in practice. However, the optimization problem LTS deals with is non-convex, for which only sub-optimal solutions are often available. Moreover, the performance of shapelets discovered by LTS highly depends on the initial shapelet, one of the input parameters. Therefore, it is also hard to give theoretical guarantees for LTS.

In this paper, we assume a possibly infinitely large set of sequences for a class of shapelets and consider them as base classifiers. We are motivated by the theoretical result that the boosting algorithm can find a good convex combinations of hypotheses from a (infinitely) large set of base classifiers. In addition, we mention that if we give the bound of empirical Rademacher complexity under a mild assumption, we can guarantee the generalization bound of shapelet-based classification. However, we have not succeeded to find efficient algorihms for the base learner. Hence, we consider an alternative approach which efficiently generates potentially effective shapelets. Our approach is to generate sufficiently many random candidates of shapelets instead of searching over a fixed class of shapelets. More precisely, this approach is based on the idea that if there exists a good shapelet associated with some sequence s, then a shapelet based on a sequence s' which is close to the sequence s is still good enough. Therefore, a sufficiently large set of random sequences could contain good shapelets even if it misses best shapelets.

The experimental results show that the proposed heuristic methods obtained accurate classifiers which are competitive to the state-of-the-art methods over the benchmark data sets with practical computation time.

2. Preliminaries

A time series is a sequence of real numbers. Our instance space is a set $\mathcal{T} \subseteq \mathbb{R}^*$ of time series. The learner receives a labeled sample $A = ((\mathbf{t}_1, y_i), \ldots, (\mathbf{t}_m, y_m)) \in (\mathcal{T} \times \{-1, 1\})^m$ and produces a binary classifier for time series in $\mathcal{T}$. We assume that each $(\mathbf{t}_i, y_i)$ is independently and identically distributed according to some unknown distribution D over $\mathcal{T} \times \{-1, 1\}$. For a "short" time series $\mathbf{s} \in \mathbb{R}^*$, we define the *base classifier* (or the *feature*) induced by $\mathbf{s}$, denoted by $h_{\mathbf{s}}$, as the function which maps a given time series $\mathbf{t}$ to the minimum over all subsequences of $\mathbf{t}$ of the Euclidean distance between $\mathbf{s}$ and the subsequence.

More precisely,

$$h_{\mathbf{s}}(\mathbf{t}) = \min_{1 \le l \le |\mathbf{t}|-|\mathbf{s}|+1} \frac{1}{|\mathbf{s}|} \left\| \mathbf{t}_{l:l+|\mathbf{s}|-1} - \mathbf{s} \right\|_2^2, \tag{1}$$

where $|\mathbf{t}|$ denotes the length of $\mathbf{t}$ and $\mathbf{t}_{l:r} = (t_l, \ldots, t_r)$ the subsequence of $\mathbf{t}$ from position l to r. Note that $h_{\mathbf{s}}$ is defined only for $\mathbf{t}$ such that $|\mathbf{t}| \ge |\mathbf{s}|$. Fix a (possibly infinitely large) set $\mathcal{S}$ of short time series, called the *base time series*, and let

$$H = \{h_{\mathbf{s}} \mid \mathbf{s} \in \mathcal{S}\}$$

be the set of *base classifiers*, where we assume $\min_{\mathbf{t} \in \mathcal{T}} |\mathbf{t}| \ge \max_{\mathbf{s} \in \mathcal{S}} |\mathbf{s}|$. We can also define a feature map $\Phi : \mathcal{T} \to \mathbb{R}^{\mathcal{S}}$ which maps a time series $\mathbf{t}$ to $\Phi(\mathbf{t}) = (h_{\mathbf{s}}(\mathbf{t}))_{\mathbf{s} \in \mathcal{S}}$, where the dimension of the feature space is $|\mathcal{S}|$ (possibly infinite).

The problem is to find a function g as the final hypothesis from the convex combinations of base classifiers in H. That is, the final classifier g is of the form of

$$g(\mathbf{t}) = \operatorname{sign}\left(\sum_{\mathbf{s} \in S} w_{\mathbf{s}} h_{\mathbf{s}}(\mathbf{t}) \right),$$

where $S \in \mathcal{S}$ is a finite support of base time series and $w_{\mathbf{s}}$ are real numbers that satisfy $w_{\mathbf{s}} \ge 0$ for every $\mathbf{s} \in S$ and $\sum_{\mathbf{s} \in S} w_{\mathbf{s}} = 1$. The goal is to find g with small generalization error:

$$\mathcal{E}_D(g) = \Pr_{(\mathbf{t}, y) \sim D} [g(\mathbf{t}) \ne y].$$

The support S is called the *shapelets*.

3. Learning Shapelets with LPBoost

3.1 The Formulation of Our Optimization Problem

It is well known that LPBoost algorithm, which is one of the boosting algorithm, is theoretically effective ensemble methods based on margin-maximizing theory. LPBoost returns a classification function f combining several base classfiers from any (possibly infinitely large) set of base classifiers. Using the formulation of LPBoost, our problem is designed as following optimization problem:

$$\begin{aligned}
\max_{\rho, \mathbf{w}, \boldsymbol{\xi}} \quad & \rho - C \sum_{i=1}^{m} \xi_i \qquad (2) \\
\text{sub.to} \quad & \sum_{\mathbf{s} \in \mathcal{S}} y_i (w_{\mathbf{s}} h_{\mathbf{s}}(\mathbf{t}_i)) \ge \rho - \xi_i, \quad i = 1, \ldots, m, \\
& \boldsymbol{\xi} \ge \mathbf{0}, \\
& \sum_{\mathbf{s} \in \mathcal{S}} w_{\mathbf{s}} = 1, \\
& \rho \in \mathbb{R},
\end{aligned}$$

where C is a constant parameter, ρ is a *target margin* and ξ_i is a slack variable. On the other hand, the dual form of the problem (2) is given as follows.

$$\begin{aligned} \min_{\gamma,\boldsymbol{\alpha}} \quad & \gamma && (3) \\ \text{sub.to} \quad & \sum_{i=1}^{m} y_i(\alpha_i h_{\mathbf{s}}(\mathbf{t}_i)) \geq \gamma, \quad \mathbf{s} \in \mathcal{S}, \\ & 0 \leq \alpha_i \leq C, \quad i = 1, \ldots, m, \\ & \sum_{i=1}^{m} \alpha_i = 1, \\ & \gamma \in \mathbb{R}. \end{aligned}$$

In many cases, such linear programming problem can be solved efficiently by using column generation algorithm.However, the shapelet-based classifier $h_{\mathbf{s}}$ requires the computation of minimization (1), and it seems to be hard to find efficient algorithms for the base learner.

3.2 Theoretical Observation

Mohri et al. (2012, Theorem 6.2 and Corollary 6.1) shows the *empirical Rademacher complexity* $\mathfrak{R}_A$ of a set of base classifier H equal to $\mathfrak{R}_A$ of its convex hull $\mathrm{conv}(H)$. Moreover, for fixed ρ, $\delta > 0$, and for all $f \in \mathrm{conv}(H)$, the following bound holds with probability at least $1 - \delta$:

$$\mathcal{E}_D(f) \leq \mathcal{E}_\rho(f) + \frac{2}{\rho}\mathfrak{R}_A(H) + 3\sqrt{\frac{\log \frac{1}{\delta}}{2m}},$$

where $\mathcal{E}_\rho$ is the *empirical margin loss* of f. Thus, if we give the bound of $\mathfrak{R}_A(H)$ under a mild assumption, we can guarantee the generalization bound of shapelet-based classification.

4. A Heuristic Approach

In practice, it is hard to implement an algorithm to solve the optimization problem 2. Thus, in this section, we show a heuristic approximate approach using random sequences that allows us to obtain good hypothesis and shapelets in practice. More precisely, we randomly choice large number N of $\mathbf{s}_1 \ldots, \mathbf{s}_N$ from $\mathcal{S}$ as possible, and get N base classifiers (features) $h_{\mathbf{s}_1}(\mathbf{t}), \ldots, h_{\mathbf{s}_N}(\mathbf{t})$. Finally, we learn a final classifier g using sparse learning model such as LPBoost or SVM with 1-norm of the weight vector regularized.

4.1 Random Sequences Generation

First, we introduce an approach to prepare finite number of base time series $\mathbf{s}_1 \ldots, \mathbf{s}_N$ from $\mathcal{S}$. The algorithm is very simple. It only generates N sequences $\mathbf{s}_1, \ldots, \mathbf{s}_N$ randomly. This approach is based on the idea that the large enough number N of random sequences might contain good shapelets. In this paper, we introduce three types of random sequence generation schema as below.

Random Choosing from Subsequences (RCS)

Input parameters: training time-series examples $\mathbf{t}_1, \ldots, \mathbf{t}_m$, a range of length of the base time series, $R_{\min}, R_{\max}$, number of the base time series N.

Iterate N times: Choose $R \in \{R_{\min}, \dots, R_{\max}\}$, $i \in \{1, \dots, m\}$ and $l \in \{1, \dots, |\mathbf{t}_i| - R + 1\}$ uniformly at random, respectively, and get a subsequence $\mathbf{s}_k = \mathrm{sub}_{l,R}(\mathbf{t}_i)$.

Random Values (RV)

Input parameters: training time-series examples $\mathbf{t}_1, \dots, \mathbf{t}_m$, a range of length of the base time series, $R_{\min}, R_{\max}$, number of the base time series N.

Iterate N times: Choose $R \in \{R_{\min}, \dots, R_{\max}\}$ uniformly at random, generate R random values according to the Gaussian distribution with mean μ and standard deviation σ,where μ and σ are the mean and the standard deviation of all the points of the time-series examples, respectively.

Random-Walk (RW)

Input parameters: training time-series examples $\mathbf{t}_1, \dots, \mathbf{t}_m$, a range of length of the base time series, $R_{\min}, R_{\max}$, number of the base time series N.

Iterate N times: Choose $R \in \{R_{\min}, \dots, R_{\max}\}$ uniformly at random, generate an initial value s_1 from Gaussian distribution with mean μ_1 and standard deviation σ_1, and generate s_r $(r = 2, \dots, R)$ given by $s_{r-1} + a_r$, where a_r is distributed from Gaussian distribution with mean μ_2 and standard deviation σ_2, where (μ_1, σ_1) are the mean and the standard deviation of all the points of the time-series examples, and (μ_2, σ_2) are the mean and the standard deviation of the all the differences of all the differences (e.g., $t_l - t_{l+1}$), respectively.

Note that RCS is different from the other random subsequence selection methods (Karlsson et al., 2016; Renard et al., 2015; Wistuba et al., 2015) in that it chooses fixed N subsequences at random just once. Since RW outputs smooth sequences compared to RV, we consider the extracted shapelets to be more interpretable.

4.2 Approximate Learning using Sparse SVM

Note that LPBoost is equal to SVM with 1-norm of the weight vector regularized, which we call SSVM (Sparse SVM). More preciely, we consider N base classifiers as N dimensional feature vectors $\mathbf{x} = (h_{\mathbf{s}_1}, \dots, h_{\mathbf{s}_N})$. Here we show a formulation of SSVM as below:

$$\begin{aligned}
&\max_{\rho, \mathbf{w}, b, \boldsymbol{\xi}} \rho - \frac{1}{\nu} \sum_{i=1}^{m} \xi_i \\
&\text{sub.to} \\
&y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq \rho - \xi_i, \quad i = 1, \dots, m, \\
&\boldsymbol{\xi} \geq \mathbf{0}, \\
&\|\mathbf{w}\|_1 = 1, \\
&\rho \geq 0,
\end{aligned}$$

where b is a bias term, $\| \cdot \|_1$ means 1-norm of $\cdot$. Here, ν is a parameter that controls the tradeoffs between two objectives, maximizing ρ and minimizing ξ_i's.

The optimization problem can be solved using a linear programming solver. SSVM finds a linear hypothesis $g(\mathbf{x}) = \mathrm{sign}(\mathbf{w} \cdot \mathbf{x} + b)$, for some weight vector $\mathbf{w} \in \mathbb{R}^N$ and real number

$b \in \mathbb{R}$. The resulting weight vector tends to be sparse and thus is suitable for feature selection. Therefore, we can choose relevant features associated with shapelets.

We would like to mention that our approach, based on a sparse learning model, can be broadly applicable to the other learning problem according to the problem we want to solve. For example, AUC (Area Under the ROC Curve) maximization problem using sparse Ranking SVMs (Suehiro et al., 2011; Narasimhan and Agarwal, 2013) or RankBoost (Freund et al., 2003), and regression problem using lasso (Tibshirani, 1994) for time series data.

5. Experiments

In the following experiments, we show that our heuristic methods are practically effective for time-series classification problem.

5.1 Classification Accuracy for UCR Datasets

We use UCR datasets (Chen et al., 2015), that are used as benchmark datasets for time-series classification methods. Table 1 shows that the detailed information of used datasets. For multi-class dataset, using one-vs-all technique, we evaluate average accuracy of multiple binary classification.

Table 1: The detail of used data sets.

dataset	m	# test	L	C
Beef	30	30	470	5
Coffee	28	28	286	2
Diatom.	16	306	345	4
ECGFiveDays	23	861	136	2
Face (four)	24	88	350	4
Gun-Point	50	150	150	2
ItalyPower.	67	1029	24	2
MoteStrain	20	1252	84	2
SonyAIBO.	20	601	70	2
Trace	100	100	275	4
TwoLeadECG	23	1139	82	2

We set the sequences-generation parameters for RCS, RV, and RW, range of subsequence length is between $R_{\min} = 0.1 \times L$ and $R_{\max} = 0.2 \times L$ (round down) through the all experiments, where L is the length of each time series dataset. We set the number of base classifiers (features) $N = m \times \{10, 100, 500\}$, respectively. The SSVM parameter ν was searched from $\nu = \{0.2, 0.4\}$. We found the best parameters set for N and ν through a grid search using 5-fold cross validation over the training data. In order to consider the randomness of our methods, we train 5 times at each parameter set and evaluate the average accuracy.

The results of classification accuracy are shown in Table 2. We referred to the experimental result of Hou et al. (2016) with regard to the accuracies of the following algorithms:

- LTS: Learning Time-Series Shapelets algorithm (Grabocka et al., 2014).

- IG-SVM: using standard (2-norm weight vector regularized) linear SVM and information-gain based shapelets search, (Hills et al., 2014).
- FLAG: The state-of-the-art algorithm proposed by Hou et al. (2016), that searches high-quality shapelets efficiently using fused-lasso (Tibshirani et al., 2005) to learn shapelet indicator vector.

We can see that our proposed methods could not defeat but comparable to the existing state-of-the-art methods.

Table 2: Classification accuracies (%) comparing with the existing methods

dataset	IG-SVM	LTS	FLAG	RCS-SVM	RV-SSVM	RW-SSVM
Beef	**90.0**	76.7	83.3	80.5	78.9	77.2
Coffee	**100.0**	**100.0**	**100.0**	95.7	92.9	**100.0**
Diatom.	93.1	94.2	**96.4**	95.3	87.4	96.3
ECGFiveDays	99.0	**100.0**	92.0	99.0	97.3	99.8
Face (four)	**97.7**	94.3	90.9	95.2	92.4	89.7
Gun-Point	**100.0**	99.6	96.7	95.5	98.1	98.3
ItalyPower.	93.7	**95.8**	94.6	88.3	88.8	89.4
MoteStrain	88.7	**90.0**	88.8	85.1	80.0	85.3
SonyAIBO.	**92.7**	91.0	92.9	91.7	83.56	89.6
Trace	98.0	**100.0**	99.0	99.8	**100.0**	**100.0**
TwoLeadECG	**100.0**	**100.0**	99.0	98.2	99.8	99.9

The next experiment compares our algorithms themselves using large number of random sequences as base time series. The number of base time series of our methods are fixed at the number of all $R_{\min}$ to $R_{\max}$ length of subsequences in training data, thus, we examine to use all of the subsequences instead of RCS. Table 3 shows that the accuracies for the above UCR datasets using large number of sequences as base time series. Interestingly, the methods based on random-valued sequences (RV and RW) and the methods based on the all of subsequences are more or less the same.

5.2 Practicality for Large Dataset

Next experiment shows the practicality of our algorithms using a large dataset. For our SSVM-based algorithms, we tried to learn shapelets and classification rules using *StarLightCurves* training data ($m = 1000$, $L = 1024$, $C = 3$). The parameter ν of SSVM is fixed at 0.2, and the number of the base time series is $N = \{100, 1000, 10000, 100000\}$, respectively. The experiments are run on the machine with 10 cores of Intel Xeon CPU 2.80GHz and 256GByte memory. We implemented our algorithms in R and used CPLEX software as the LP solver for SSVM. Table 4 shows that classification accuracy, total computation time for sequences generation and learning with SSVM, when using each N base time series.

We can observe that the classification accuracy gets higher and the number of extracted shapelets are increasing with increasing N. Actually, almost of the computation time is consumed by calculating feature vectors. The time complexity of calculating feature vectors is $O(mNL)$, thus, linearly increasing with N. We cannot compare with the result of existing

Table 3: Classification accuracies (%) for our methods using large number of sequences as of base time series.

dataset	N	RC(Full-subsequences)-SSVM	RV-SSVM	RW-SSVM
Beef	576720	79.3	**79.9**	76.7
Coffee	205380	96.4	97.1	**100.0**
Diatom.	169632	95.4	90.3	**96.8**
ECGFiveDays	40365	98.7	97.3	**100.0**
Face (Four)	257904	**95.1**	93.4	90.1
Gun-Point	102800	98.0	97.6	**99.1**
ItalyPower.	4422	85.0	**90.9**	90.5
MoteStrain	13140	**85.5**	79.7	83.4
SonyAIBO.	9680	**93.8**	83.7	89.2
Trace	681500	99.5	91.2	**100.0**
TwoLeadECG	14697	98.9	99.4	**99.9**

Table 4: Performance of our algorithms for large dataset with increasing N.

N	accuracy	computation time(sec.)	#shapelets
100	79.1	84.0	16
1000	89.5	279.2	20
10000	90.8	1950.1	36
100000	92.1	18983.8	44

methods since experiment environments are different. However, we can mention that our methods work practically when using large dataset.

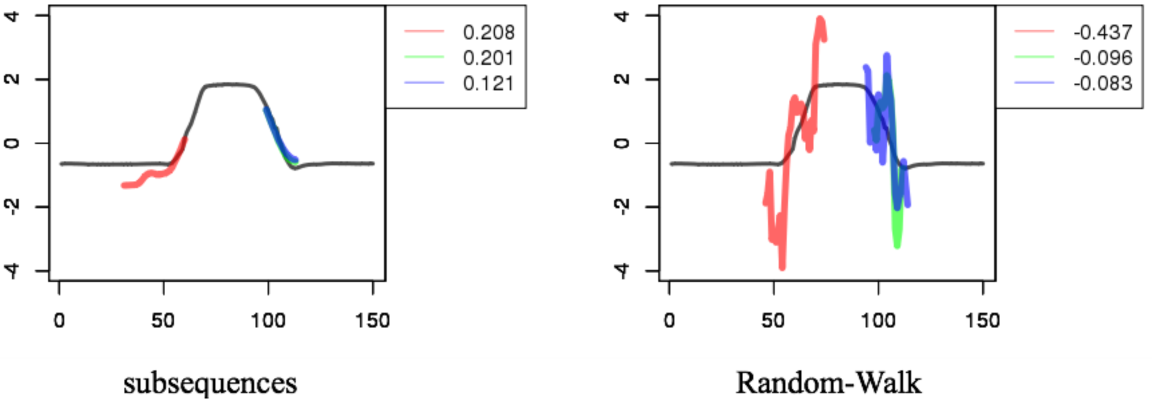
5.3 Efficiency of Sparseness and Interpretation

Finally, we show that the weight vector obtained by SSVM has high sparsity which allow us to interpret the important shapelets. Table 5 shows that number of shapelets obtained by SSVM and standard linear SVM (2-norm weight vector regularized), where $R_{\min} = 15$ and $R_{\max} = 30$, $K = \{500, 5000, 25000, 102800 \ (= \#\text{subsequences})\}$, the base time series of shapelet are generated by RW. The number of shapelets obtained by SSVM keeps little more than 10 when increasing the number of the base time series N, whereas standard SVM always weights all of the shapelets. Moreover, 2-norm regularized SVM does not work at $N = 102800$, since 2-norm regularized SVM needs to solve QP problem, which costs large memory and computation time naively.

We show the examples of shapelets obtained by our algorithms, SSVM using the full of subsequences and RW of $N = 25000$ in Figure 1. It is an interesting result that the extracted shapelets by SSVM with RW are roughly similar to SSVM with full of subsequences.

Table 5: The number of shapelets obtained by SSVM and standard linear SVM for Gun-Point training data.

N	SSVM	2-norm reg. SVM
500	10	500
5000	13	5000
25000	12	25000
102800	12	out of memory.



subsequences

Random-Walk

Figure 1: An illustration of the extracted shapelets by our two methods (based on all of subsequences and RW) for an example of Gun-point data (label: 2). The colored lines are top 3 highly weighted shapelets. The legend of each plot means the weight of each shapelet.

6. Discussion

The experimental results show that our proposed methods are comparable to the state-of-the-art methods while reducing computational cost significantly. In order to improve accuracies, efficiency and to extract more informative shapelets, we need to verify our approach on more various data sets using much more parameters, and consider other heuristics to generate random sequences. However, a better goal would be to provide a theoretical guarantee for shapelet-based time series classification. First, we would like to give the bound of the empirical Rademacher complexity of $\mathfrak{R}_A(h_\mathbf{s})$. Second, we are willing to consider the efficient algorithms to solve the optimization problem (2) and give the (approximate) guarantee. Of course, our proposed heuristic approach might have theoretical guarantee. Balcan et al. (2008) have shown that if there exists a large margin hyperplane in the feature space using any similarity function between instances, then large enough instances (randomly drawn from an input distribution) can still construct the feature space which is almost linearly separable. Their analyses can be generalized to our case where the feature space is defined not by similarity measures between instances but by similarity measures between shapelets and instances. In addition, we consider the way to kernelize as one of the best possible approach for efficiency.

7. Conclusion and Future Work

In this paper, we consider a time series classification problem using shapelets. The proposed novel approach, which is based on random sequences and sparse modeling, performs practically accurate. As a future work, we would like to bound the empirical Rademacher complexity of shapelet-based hypothesis class, and consider effective algorithms to obtain a final classifier.

References

Maria-Florina Balcan, Avrim Blum, and Nathan Srebro. Improved Guarantees for Learning via Similarity Functions. In *Proceedings of 21st Annual Conference on Learning Theory (COLT2008)*, pages 287–298, 2008.

Yanping Chen, Eamonn Keogh, Bing Hu, Nurjahan Begum, Anthony Bagnall, Abdullah Mueen, and Gustavo Batista. The ucr time series classification archive, July 2015. `www.cs.ucr.edu/~eamonn/time_series_data/`.

Yoav Freund, Raj Iyer, Robert E. Shapire, and Yoram Singer. An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, 4:933–969, 2003.

Josif Grabocka, Nicolas Schilling, Martin Wistuba, and Lars Schmidt-Thieme. Learning time-series shapelets. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pages 392–401, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2956-9.

Josif Grabocka, Martin Wistuba, and Lars Schmidt-Thieme. Scalable discovery of time-series shapelets. *CoRR*, abs/1503.03238, 2015. URL `http://arxiv.org/abs/1503.03238`.

Jon Hills, Jason Lines, Edgaras Baranauskas, James Mapp, and Anthony Bagnall. Classification of time series by shapelet transformation. *Data Min. Knowl. Discov.*, 28(4): 851–881, July 2014. ISSN 1384-5810.

Lu Hou, James T. Kwok, and Jacek M. Zurada. Efficient learning of timeseries shapelets. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*, pages 1209–1215, 2016.

Isak Karlsson, Panagiotis Papapetrou, and Henrik Boström. Generalized random shapelet forests. *Data Min. Knowl. Discov.*, 30(5):1053–1085, September 2016. ISSN 1384-5810.

Eamonn J. Keogh and Thanawin Rakthanmanon. Fast shapelets: A scalable algorithm for discovering time series shapelets. In *Proceedings of the 13th SIAM International Conference on Data Mining, May 2-4, 2013. Austin, Texas, USA.*, pages 668–676, 2013.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. The MIT Press, 2012.

Harikrishna Narasimhan and Shivani Agarwal. Svmpauctight: A new support vector method for optimizing partial auc based on a tight convex upper bound. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, pages 167–175, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2174-7.

Xavier Renard, Maria Rifqi, Walid Erray, and Marcin Detyniecki. Random-shapelet: an algorithm for fast shapelet discovery. In *2015 IEEE International Conference on Data Science and Advanced Analytics (IEEE DSAA'2015)*, pages 1–10, Paris, France, October 2015. IEEE.

Daiki Suehiro, Kohei Hatano, and Eiji Takimoto. Approximate reduction from AUC maximization to 1-norm soft margin optimization. In *Proceedings of the 22nd International Conference on Algorithmic Learning Theory*, pages 324–337, 2011.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1994.

Robert Tibshirani, Michael Saunders, Saharon Rosset, Ji Zhu, and Keith Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society Series B*, pages 91–108, 2005.

Martin Wistuba, Josif Grabocka, and Lars Schmidt-Thieme. Ultra-fast shapelets for time series classification. *CoRR*, abs/1503.05018, 2015. URL `http://arxiv.org/abs/1503.05018`.

Lexiang Ye and Eamonn Keogh. Time series shapelets: A new primitive for data mining. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 947–956, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-495-9.