LETTER Communicated by Ghyslain Gagnon

Theory and Algorithms for Shapelet-Based Multiple-Instance Learning

Daiki Suehiro
suehiro93@gmail.com
Department of Advanced Information Technology, Faculty of Information Science and Electrical Engineering, Kyushu University, and RIKEN Center for Advanced Intelligence Project, Nishi-ku, Fukuoka, 8190395, Japan

Kohei Hatano
hatano@inf.kyushu-u.ac.jp
Faculty of Arts and Science, Kyushu University, and RIKEN Center for Advanced Intelligence Project, Nishi-ku, Fukuoka, 8190395, Japan

Eiji Takimoto
eiji@inf.kyushu-u.ac.jp
Department of Informatics, Faculty of Information Science and Electrical Engineering, Kyushu University, Nishi-ku, Fukuoka, 8190395, Japan

Shuji Yamamoto
yamashu@math.keio.ac.jp
Kenichi Bannai
bannai@math.keio.ac.jp
Department of Mathematics, Keio University, and RIKEN Center for Advanced Intelligence Project, Minatokita-ku, Yokohama, 2238522, Japan

Akiko Takeda
takeda@mist.i.u-tokyo.ac.jp
Department of Creative Informatics, University of Tokyo, and RIKEN Center for Advanced Intelligence Project, Bunkyo-ku, Tokyo, 1138656, Japan

We propose a new formulation of multiple-instance learning (MIL), in which a unit of data consists of a set of instances called a bag. The goal is to find a good classifier of bags based on the similarity with a "shapelet" (or pattern), where the similarity of a bag with a shapelet is the maximum similarity of instances in the bag. In previous work, some of the training instances have been chosen as shapelets with no theoretical justification. In our formulation, we use all possible, and thus infinitely many, shapelets, resulting in a richer class of classifiers. We show that the formulation is tractable, that is, it can be reduced through linear programming boosting (LPBoost) to difference of convex (DC) programs of finite

Neural Computation **32, 1580–1613** (2020)
https://doi.org/10.1162/neco_a_01297

(actually polynomial) size. Our theoretical result also gives justification to the heuristics of some previous work. The time complexity of the proposed algorithm highly depends on the size of the set of all instances in the training sample. To apply to the data containing a large number of instances, we also propose a heuristic option of the algorithm without the loss of the theoretical guarantee. Our empirical study demonstrates that our algorithm uniformly works for shapelet learning tasks on time-series classification and various MIL tasks with comparable accuracy to the existing methods. Moreover, we show that the proposed heuristics allow us to achieve the result in reasonable computational time.

1 Introduction

Multiple-instance learning (MIL) is a fundamental framework of supervised learning with a wide range of applications, such as prediction of molecular activity, and image classification. It has been extensively studied in both theoretical and work (Gärtner et al., 2002; Andrews, Tsochantaridis, & Hofmann, 2003; Sabato & Tishby, 2012; Zhang, He, Si, & Lawrence, 2013; Doran & Ray, 2014; Carbonneau, Cheplygina, Granger, & Gagnon, 2018), since the notion of MIL was first proposed by Dietterich, Lathrop, and Lozano-Pérez (1997).

A standard MIL setting is described as follows: A learner receives sets $B_1, B_2, \ldots, B_m$ called *bags*; each contains multiple instances. In the training phase, each bag is labeled, but instances are not labeled individually. The goal of the learner is to obtain a hypothesis that predicts the labels of unseen bags correctly.[1] One of the most common hypotheses used in practice has the following form,

$$h_{\mathbf{u}}(B) = \max_{x \in B} \left\langle \mathbf{u}, \Phi(x) \right\rangle, \tag{1.1}$$

where Φ is a feature map and $\mathbf{u}$ is a feature vector that we call a *shapelet*. In many applications, $\mathbf{u}$ is interpreted as a particular pattern in the feature space and the inner product as the similarity of $\Phi(x)$ from $\mathbf{u}$. Note that we use the term *shapelets* by following the terminology of shapelet learning (SL), which is a framework for time-series classification, although it is often called *concepts* in the literature of MIL. Intuitively, this hypothesis evaluates a given bag by the maximum similarity among the instances in the bag and the shapelet $\mathbf{u}$. The multiple-instance support vector machine (MI-SVM), proposed by Andrews et al. (2003), is a widely used algorithm that employs this hypothesis class and learns $\mathbf{u}$. It is well known that MIL algorithms using this hypothesis class perform empirically better in various

[1] Although there are settings where instance label prediction is also considered, we focus only on bag-label prediction in this letter.

multiple-instance data sets. Moreover, a generalization error bound of the hypothesis class is given by Sabato and Tishby (2012).

However, in some domains, such as image recognition and document classification, it is said that the hypothesis class 1.1 is not effective (see, e.g., Chen, Bi, & Wang, 2006). To employ MIL on such domains more effectively, Chen et al. (2006) extend a hypothesis to a convex combination of $h_{\mathbf{u}}$,

$$g(B) = \sum_{\mathbf{u} \in U} w_{\mathbf{u}} \max_{x \in B} \left\langle \mathbf{u}, \Phi(x) \right\rangle, \tag{1.2}$$

for some set U of shapelets. In particular, Chen et al. (2006) consider $U_{\text{train}} = \{\Phi(z) \mid z \in \bigcup_{i=1}^{m} B_i\}$, which is constructed from all instances in the training sample. They demonstrate that this hypothesis with the gaussian kernel performs well in image recognition. The generalization bound provided by Sabato and Tishby (2012) is applicable to a hypothesis class of the form 1.2 for the set U of infinitely many shapelets $\mathbf{u}$ with bounded norm. Therefore, the generalization bound also holds for U_{train}. However, it has never been theoretically discussed why such a fixed set U_{train} using training instances effectively works in MIL tasks.

1.1 Our Contributions. In this letter, we propose an MIL formulation with the hypothesis class 1.2 for sets U of infinitely many shapelets.

The proposed learning framework is theoretically motivated and practically effective. We show the generalization error bound based on the Rademacher complexity (Bartlett & Mendelson, 2003) and large margin theory. The result indicates that we can achieve a small generalization error by keeping a large margin for a large training sample.

The learning framework can be applied to various kinds of data and tasks because of our unified formulation. The existing shapelet-based methods are formulated for their target domains. More precisely, the existing shapelet-based methods are formulated using a fixed similarity measure (or distance), and the generalization ability is shown empirically in their target domains. For example, Chen et al. (2006) and Sangnier, Gauthier, and Rakotomamonjy (2016) calculated the feature vectors based on the similarity between every instance using the gaussian kernel. In the time-series domain, shapelet-based methods (Ye & Keogh, 2009; Keogh & Rakthanmanon, 2013; Hills et al., 2014) usually use Euclidean distance as a similarity measure (or distance). By contrast, our framework employs a kernel function as a similarity measure. Therefore, our learning framework can be uniformly applied if we can set a kernel function as a similarity measure according to a target learning—for example, the gaussian kernel (which behaves like the Euclidean distance) and dynamic time warping (DTW) kernel (Shimodaira, Noma, Nakai, & Sagayama, 2001). Our framework can be also applied to non-real-valued sequence data (e.g., text and a discrete signal) using a string kernel. Moreover, our generalization performance is guaranteed theoretically. The experimental results demonstrate that the approach uniformly works for SL and MIL tasks without introducing domain-specific parameters and heuristics, and it compares with the state-of-the-art shapelet-based methods.

We show that the formulation is tractable. The algorithm is based on linear programming boosting (LPBoost; Demiriz, Bennett, & Shawe-Taylor, 2002), which solves the soft margin optimization problem via a column generation approach. Although the weak learning problem in the boosting becomes an optimization problem over an infinite-dimensional space, we can show that an analog of the representer theorem holds on it and allows us to reduce it to a nonconvex optimization problem (difference of convex program) over a finite-dimensional space. While it is difficult to solve the subproblems exactly because of nonconvexity, it is possible to find good approximate solutions with in reasonable time in many practical cases (Le Thi & Pham Dinh, 2018).

Remarkably, our theoretical result gives justification to the heuristics of choosing the shapelets in the training instances. Our representer theorem indicates that at the tth iteration of boosting, the optimal solution $\mathbf{u}_t$ (i.e., shapelet) of the weak learning problem can be written as a linear combination of the feature maps of training instances, that is, $\mathbf{u}_t = \sum_{z \in \bigcup_{i=1}^{m} B_i} \alpha_{t,z} \Phi(z)$. Thus, we obtain a final classifier of the following form:

$$g(B) = \sum_{t=1}^{T} w_t \max_{x \in B} \langle \mathbf{u}_t, \Phi(x) \rangle = \sum_{t=1}^{T} w_t \max_{x \in B} \sum_{z \in \bigcup_{i=1}^{m} B_i} \alpha_{t,z} \langle \Phi(z), \Phi(x) \rangle.$$

Note that the hypothesis class used in the standard approach (Chen et al., 2006; Sangnier et al., 2016) corresponds to the special case where $\mathbf{u}_t \in U_{\text{train}} = \{\Phi(z) \mid z \in \bigcup_{i=1}^{m} B_i\}$. This observation would suggest that the standard approach of using U_{train} is reasonable.

1.2 Comparison to Related Work for MIL.

There are many MIL algorithms with hypothesis classes that are different from equations 1.1 or 1.2. (e.g., Auer & Ortner, 2004; Gärtner et al., 2002; Andrews & Hofmann, 2004; Zhang, Platt, & Viola, 2006; Chen et al., 2006). For example, these algorithms adopted diverse approaches for the bag-labeling hypothesis from shapelet-based hypothesis classes (e.g., Zhang et al., 2006, used a noisy-OR based hypothesis and Gärtner et al., 2002, proposed a new kernel called a *set kernel*). Shapelet-based hypothesis classes have a practical advantage of being applicable to SL in the time-series domain (see section 1.3).

Sabato and Tishby (2012) proved generalization bounds of hypotheses classes for MIL including those of equations 1.1 and 1.2 with infinitely large sets U. The generalization bound we provid in this letter is incomparable to the bound provided by Sabato and Tishby. When some data-dependent parameter is regarded as a constant, our bound is slightly better in terms of the sample size m by the factor of $O(\log m)$. They also proved the PAC learnability of the class 1.1 using the boosting approach under some technical assumptions. Their boosting approach is different from our work in that they assume that labels are consistent with some hypothesis of the form 1.1, while we consider arbitrary distributions over bags and labels.

1.3 Connection between MIL and Shapelet Learning for Time-Series Classification. Here we mention briefly that MIL with type 1.2 hypotheses is closely related to SL, a framework for time-series classification that has been extensively studied (Ye & Keogh, 2009; Keogh & Rakthanmanon, 2013; Hills, Lines, Baranauskas, Mapp, & Bagnall, 2014; Grabocka, Schilling, Wistuba, & Schmidt-Thieme, 2014) in parallel to MIL. SL is a notion of learning with a feature extraction method, defined by a finite set $M \subseteq \mathbb{R}^\ell$ of real-valued "short" sequences called shapelets. A similarity measure is given by (not necessarily a Mercer kernel) $K : \mathbb{R}^\ell \times \mathbb{R}^\ell \to \mathbb{R}$ in the following way. A time series $\boldsymbol{\tau} = (\tau[1], \ldots, \tau[L]) \in \mathbb{R}^L$ can be identified with a bag $B_{\boldsymbol{\tau}} = \{(\tau[j], \ldots, \tau[j + \ell - 1]) \mid 1 \leq j \leq L - \ell + 1\}$ consisting of all subsequences of $\boldsymbol{\tau}$ of length ℓ. The feature of $\boldsymbol{\tau}$ is a vector $(\max_{\mathbf{x} \in B_\tau} K(\mathbf{z}, \mathbf{x}))_{\mathbf{z} \in M}$ of a fixed dimension $|M|$ regardless of the length L of the time series $\boldsymbol{\tau}$. When we employ a linear classifier on top of the features, we obtain a hypothesis in the form

$$g(\boldsymbol{\tau}) = \sum_{\mathbf{z} \in M} w_{\mathbf{z}} \max_{\mathbf{x} \in B_\tau} K(\mathbf{z}, \mathbf{x}), \tag{1.3}$$

which is essentially the same form as equation 1.2, except that finding good shapelets M is a part of the learning task, as well as to find a good weight vector $\mathbf{w}$. This task is one of the most successful approaches for SL (Hills et al., 2014; Grabocka et al., 2014, 2015; Renard, Rifqi, Erray, & Detyniecki, 2015; Hou, Kwok, & Zurada, 2016), where a typical choice of K is $K(\mathbf{z}, \mathbf{x}) = -\|\mathbf{z} - \mathbf{x}\|_2$. However, almost all existing methods heuristically choose shapelets M and with no theoretical guarantee on how good the choice of M is.

Note also that in the SL framework, each $\mathbf{z} \in M$ is called a shapelet, while in this letter, we assume that K is a kernel $K(z, x) = \langle \Phi(z), \Phi(x) \rangle$ and any $\mathbf{u}$ (not necessarily $\Phi(z)$ for some z) in the Hilbert space is called a shapelet.

Sangnier et al. (2016) proposed an MIL-based anomaly detection algorithm for time-series data. They showed an algorithm based on LPBoost and the generalization error bound based on the Rademacher complexity (Bartlett & Mendelson, 2003). Their hypothesis class is same as the of Chen et al. (2006). However, they did not analyze the theoretical justification to use finite set U made from training instances (however, they mentioned it as future work). By contrast, we consider a hypothesis class based on infinitely many shapelets, and our representer theorem guarantees that our learning problem over the infinitely large set is still tractable. As a result, our study justifies the previous heuristics of their approach.

There is another work that treats shapelets not appearing in the training set. The learning time-series shapelets (LTS) algorithm (Grabocka et al., 2014) tries to solve a nonconvex optimization problem of learning effective shapelets in an infinitely large domain. However, there is no theoretical guarantee of its generalization error. In fact, our generalization error bound applies to their hypothesis class.

For SL tasks, many researchers focus on improving efficiency (Keogh & Rakthanmanon, 2013; Renard et al., 2015; Grabocka, Wistuba, & Schmidt-Thieme, 2015; Wistuba, Grabocka, & Schmidt-Thieme, 2015; Hou et al., 2016; Karlsson, Papapetrou, & Boström, 2016). However, these methods are specialized in the time-series domain, and the generalization performance has never been theoretically discussed.

Curiously, although MIL and SL share similar motivations and hypotheses, the relationship between them has not yet been pointed out. From the shapelet-perspective in MIL, hypothesis 1.1 is regarded as a "single shapelet"–based hypothesis, and hypothesis 1.2 is regarded as a "multiple-shapelets"–based hypothesis. In this study, we refer to a linear combination of maximum similarities based on shapelets such as equations 1.2 and (1.3) as *shapelet-based classifiers*.

2 Preliminaries

Let $\mathcal{X}$ be an instance space. A bag B is a finite set of instances chosen from $\mathcal{X}$. The learner receives a sequence of labeled bags $S = ((B_1, y_1), \ldots, (B_m, y_m)) \in (2^{\mathcal{X}} \times \{-1, 1\})^m$ called a *sample*, where each labeled bag is independently drawn according to some unknown distribution D over $2^{\mathcal{X}} \times \{-1, 1\}$. Let P_S denote the set of all instances that appear in the sample S. That is, $P_S = \bigcup_{i=1}^m B_i$. Let K be a kernel over $\mathcal{X}$, which is used to measure the similarity between instances, and let $\Phi : \mathcal{X} \to \mathbb{H}$ denote a feature map associated with the kernel K for a Hilbert space $\mathbb{H}$, that is, $K(z, z') = \langle \Phi(z), \Phi(z') \rangle$ for instances $z, z' \in \mathcal{X}$, where $\langle \cdot, \cdot \rangle$ denotes the inner product over $\mathbb{H}$. The norm induced by the inner product is denoted by $\| \cdot \|_{\mathbb{H}}$ defined as $\|\mathbf{u}\|_{\mathbb{H}} = \sqrt{\langle \mathbf{u}, \mathbf{u} \rangle}$ for $\mathbf{u} \in \mathbb{H}$.

For each $\mathbf{u} \in \mathbb{H}$ which we call a shapelet, we define a *shapelet-based classifier* denoted by $h_{\mathbf{u}}$, as the function that maps a given bag B to the maximum of the similarity scores between shapelet $\mathbf{u}$ and $\Phi(x)$ over all instances x in B. More specifically,

$$h_{\mathbf{u}}(B) = \max_{x \in B} \left\langle \mathbf{u}, \Phi(x) \right\rangle .$$

For a set $U \subseteq \mathbb{H}$, we define the class of shapelet-based classifiers as

$$H_U = \{h_{\mathbf{u}} \mid \mathbf{u} \in U\}$$

and let conv(H_U) denote the set of convex combinations of shapelet-based classifiers in H_U. More precisely,

$$\begin{aligned}\mathrm{conv}(H_U) &= \left\{ \int_{\mathbf{u}\in U} w_{\mathbf{u}} h_{\mathbf{u}} d\mathbf{u} \mid w_{\mathbf{u}} \text{ is a density over } U \right\} \\ &= \left\{ \sum_{\mathbf{u}\in U'} w_{\mathbf{u}} h_{\mathbf{u}} \mid \forall \mathbf{u} \in U', w_{\mathbf{u}} \geq 0, \right. \\ &\qquad \left. \sum_{\mathbf{u}\in U'} w_{\mathbf{u}} = 1, U' \subseteq U \text{ is a finite support} \right\}. \qquad (2.1)\end{aligned}$$

The goal of the learner is to find a hypothesis $g \in \mathrm{conv}(H_U)$, so that its generalization error $\mathcal{E}_D(g) = \Pr_{(B,y)\sim D}[\mathrm{sign}(g(B)) \neq y]$ is small. Note that since the final hypothesis sign $\circ\, g$ is invariant to any scaling of g, we assume without loss of generality that

$$U = \{\mathbf{u} \in \mathbb{H} \mid \|\mathbf{u}\|_{\mathbb{H}} \leq 1\}.$$

Let $\mathcal{E}_\rho(g)$ denote the *empirical margin loss* of g over S, that is, $\mathcal{E}_\rho(g) = |\{i \mid y_i g(B_i) < \rho\}|/m$.

3 Optimization Problem Formulation

In this letter, we formulate the problem as soft margin maximization with 1-norm regularization, which ensures a generalization bound for the final hypothesis (see, e.g., Demiriz et al., 2002). Specifically, the problem is formulated as a linear programming problem (over infinitely many variables) as follows:

$$\max_{\rho, w, \mathbf{x}i} \quad \rho - \frac{1}{\nu m} \sum_{i=1}^{m} \xi_i \qquad (3.1)$$

$$\begin{aligned}\text{sub.to} \quad & \int_{\mathbf{u}\in U} y_i w_{\mathbf{u}} h_{\mathbf{u}}(B_i) d\mathbf{u} \geq \rho - \xi_i \wedge \xi_i \geq 0,\ i \in [m], \\ & \int_{\mathbf{u}\in U} w_{\mathbf{u}} d\mathbf{u} = 1, w_{\mathbf{u}} \geq 0,\ \rho \in \mathbb{R},\end{aligned}$$

where $\nu \in [0, 1]$ is a parameter. To avoid the integral over the Hilbert space, it is convenient to consider the dual form:

$$\begin{aligned} \min_{\gamma, \mathbf{d}} \quad & \gamma \\ \text{sub.to} \quad & \sum_{i=1}^{m} y_i d_i h_{\mathbf{u}}(B_i) \leq \gamma, \ \ \mathbf{u} \in U, \\ & 0 \leq d_i \leq 1/(\nu m), \ \ i \in [m], \\ & \sum_{i=1}^{m} d_i = 1, \gamma \in \mathbb{R}. \end{aligned} \tag{3.2}$$

The dual problem is categorized as a semi-infinite program because it contains infinitely many constraints. Note that the duality gap is zero because problem 3.2 is linear and the optimum is finite (see theorem 2.2 of Shapiro, 2009). We employ column generation to solve the dual problem: solve equation 3.2 for a finite subset $U' \subseteq U$, find $\mathbf{u}$ to which the corresponding constraint is maximally violated by the current solution (*column generation part*), and repeat the procedure with $U' = U' \cup \{\mathbf{u}\}$ until a certain stopping criterion is met. In particular, we use LPBoost (Demiriz et al., 2002), a well-known and practically fast algorithm of column generation. Since the solution $\mathbf{w}$ is expected to be sparse due to the 1-norm regularization, the number of iterations is expected to be small.

Following the boosting terminology, we refer to the column generation part as weak learning. In our case, weak learning is formulated following the optimization problem:

$$\max_{\mathbf{u} \in \mathbb{H}} \sum_{i=1}^{m} y_i d_i \max_{x \in B_i} \left\langle \mathbf{u}, \Phi\left(x\right) \right\rangle \ \text{sub.to} \ \|\mathbf{u}\|_{\mathbb{H}}^2 \leq 1. \tag{3.3}$$

Thus, we need to design a weak learner for solving equation 3.3 for a given sample weighted by $\mathbf{d}$. However, it seems to be impossible to solve it directly because we have access to U only through the associated kernel. Fortunately, we prove a version of representer theorem given below, which makes equation 3.3 tractable.

Theorem 1 *(Representer Theorem). The solution* $\mathbf{u}^*$ *of equation 3.3 can be written as* $\mathbf{u}^* = \sum_{z \in P_S} \alpha_z \Phi(z)$ *for some real numbers* α_z*.*

Our theorem can be derived from a nontrivial application of the standard representer theorem (see, e.g., Mohri, Rostamizadeh, & Talwalkar, 2012). Intuitively, we prove the theorem by decomposing the optimization problem 3.3 into a number of subproblems, so that the standard representer theorem can be applied to each of the subproblems. The details of the proof are given in appendix A.

This result gives justification to the simple heuristics in the standard approach: choosing the shapelets based on the training instances. More precisely, the hypothesis class used in the standard approach (Chen et al., 2006; Sangnier et al., 2016) corresponds to the special case where $\mathbf{u} \in U_{\text{train}} = \{\Phi(z) \mid z \in P_S\}$. Thus, our representer theorem would suggest that the standard approach of using U_{train} is reasonable.

Theorem 1 says that the weak learning problem can be rewritten in the following tractable form:

OP 1. Weak Learning Problem

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & -\sum_{i=1}^{m} d_i y_i \max_{x \in B_i} \sum_{z \in P_S} \alpha_z K(z, x) \\ \text{sub.to} \quad & \sum_{z \in P_S} \sum_{v \in P_S} \alpha_z \alpha_v K(z, v) \leq 1. \end{aligned}$$

Unlike the primal solution $\mathbf{w}$, the dual solution $\boldsymbol{\alpha}$ is not expected to be sparse. To obtain a more interpretable hypothesis, we propose another formulation of weak learning where 1-norm regularization is imposed on $\boldsymbol{\alpha}$, so that a sparse solution of $\boldsymbol{\alpha}$ will be obtained. In other words, instead of U, we consider the feasible set $\hat{U} = \left\{\sum_{z \in P_S} \alpha_z \Phi(z) : \|\boldsymbol{\alpha}\|_1 \leq 1\right\}$, where $\|\boldsymbol{\alpha}\|_1$ is the 1-norm of $\boldsymbol{\alpha}$.

OP 2. Sparse Weak Learning Problem

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & -\sum_{i=1}^{m} d_i y_i \max_{x \in B_i} \sum_{z \in P_S} \alpha_z K(z, x) \\ \text{sub.to} \quad & \|\boldsymbol{\alpha}\|_1 \leq 1 \end{aligned}$$

Note that when running LPBoost with a weak learner for OP 2, we obtain a final hypothesis that has the same form of generalization bound as the one stated in theorem 2, which is of a final hypothesis obtained when used with a weak learner for OP 1. To see this, consider a feasible space $\hat{U}_\Lambda = \left\{\sum_{z \in P_S} \alpha_z \Phi(z) : \|\boldsymbol{\alpha}\|_1 \leq \Lambda\right\}$ for a sufficiently small $\Lambda > 0$, so that $\hat{U}_\Lambda \subseteq U$. Then, since $H_{\hat{U}_\Lambda} \subseteq H_U$, a generalization bound for H_U also applies to $H_{\hat{U}_\Lambda}$. On the other hand, since the final hypothesis $\text{sign} \circ g$ for $g \in \text{conv}(H_{\hat{U}_\Lambda})$ is invariant to the scaling factor Λ, the generalization ability is independent of Λ.

4 Algorithms

In this section, we present the pseudocode of LPBoost in algorithm 1 for completeness. Moreover, we describe our algorithms for the weak learners.

Algorithm 1: LPBoost Using WeakLearn.

```
Inputs:
    $S$, kernel $K$, $\nu \in (0, 1]$, $\epsilon > 0$
Initialize:
    $\mathbf{d}_0 \leftarrow (\frac{1}{m}, \dots, \frac{1}{m}), \gamma = 0$
for $t = 1, \dots$ do
    $h_t \leftarrow$ Run WeakLearn($S$, $K$, $\mathbf{d}_{t-1}$, $\epsilon$)
    if $\sum_{i=1}^{m} y_i d_i h_t(B_i) \le \gamma$ then
        $t = t - 1$, break
    end if
    $(\gamma, \mathbf{d}_t) \leftarrow \arg\min_{\gamma, \mathbf{d}} \ \gamma$
    sub.to $\sum_{i=1}^{m} y_i d_i h_j(B_i) \le \gamma \ \ (j = 1, \dots, t),$
    $0 \le d_i \le 1/\nu m \ \ (i \in [m]), \sum_{i=1}^{m} d_i = 1, \ \gamma \in \mathbb{R}.$
end for
$\mathbf{w} \leftarrow$ Lagrangian multipliers of the last solution
$g \leftarrow \sum_{j=1}^{t} w_j h_j$
return $\mathrm{sign}(g)$
```

For simplicity, we denote by $\mathbf{k}_x \in \mathbb{R}^{P_S}$ a vector given by $k_{x,z} = K(z, x)$ for every $z \in P_S$. The objective function of OP 1 (and OP 2) is rewritten as

$$\sum_{i:y_i=-1} d_i \max_{x \in B_i} \mathbf{k}_x^T \boldsymbol{\alpha} - \sum_{i:y_i=1} d_i \max_{x \in B_i} \mathbf{k}_x^T \boldsymbol{\alpha},$$

which can be seen as a difference $F - G$ of two convex functions F and G of $\boldsymbol{\alpha}$. Therefore, the weak learning problems are DC programs, and thus we can use the DC algorithm (Tao & Souad, 1988; Yu & Joachims, 2009) to find an ϵ-approximation of a local optimum. We employ a standard DC algorithm. That is, for each iteration t, we linearize the concave term G with $\nabla_{\boldsymbol{\alpha}} G(\boldsymbol{\alpha}_t)^T \boldsymbol{\alpha}$ at the current solution $\boldsymbol{\alpha}_t$, which is $\sum_{i:y_i=1} d_i \mathbf{k}_{x_i^*}^T \boldsymbol{\alpha}$ with $x_i^* = \arg\max_{x \in B_i} \mathbf{k}_x^T \boldsymbol{\alpha}$ in our case, and then update the solution to $\boldsymbol{\alpha}_{t+1}$ by solving the resultant convex optimization problem OP'_t.

In addition, the problems OP'_t for OP 1 and OP 2 are reformulated as a second-order cone programming (SOCP) problem and an LP problem, respectively, and thus both problems can be efficiently solved. To this end, we introduce new variables λ_i for all negative bags B_i with $y_i = -1$, which represent the factors $\max_{x \in B_i} \mathbf{k}_x^T \boldsymbol{\alpha}$. Then we obtain the equivalent problem

Algorithm 2: WeakLearn Using the DC Algorithm.

```
Inputs:
    $S$, $K$, $\mathbf{d}$, $\epsilon$ (convergence parameter)
Initialize:
    $\boldsymbol{\alpha}_0 \in \mathbb{R}^{|P_S|}$, $f_0 \leftarrow \infty$
for $t = 1, \dots$ do
    for $\forall k : y_k = +1$ do
        $x_k^* \leftarrow \arg\max_{x \in B_k} \sum_{z \in P_S} d_k \alpha_{t,z} K(z, x)$
    end for
    $f \leftarrow \min_{\boldsymbol{\alpha}, \boldsymbol{\lambda}} - \sum_{k: y_k = +1} d_k \sum_{z \in P_S} \alpha_z K(z, x_k^*) + \sum_{r: y_r = -1} d_r \lambda_r$   (5.1)
      sub.to $\sum_{z \in P_S} \alpha_z K(z, x) \leq \lambda_r \ (\forall r : y_r = -1, \forall x \in B_r)$,
             $\sum_{z \in P_S} \sum_{v \in P_S} \alpha_z \alpha_v K(z, v) \leq 1$.
    $\boldsymbol{\alpha}_t \leftarrow \boldsymbol{\alpha}$, $f_t \leftarrow f$
    if $f_{t-1} - f_t \leq \epsilon$ then
        break
    end if
end for
return $h(B) = \max_{x \in B} \sum_{z \in P_S} \alpha_{t,z} K(z, x)$
```

to OP'_t for OP 1 as follows:

$$\min_{\boldsymbol{\alpha}, \boldsymbol{\lambda}} \quad \sum_{i: y_i = -1} d_i \lambda_i - \sum_{i: y_i = 1} d_i \max_{x \in B_i} \mathbf{k}_x^T \boldsymbol{\alpha} \tag{4.1}$$

$$\text{sub.to} \quad \mathbf{k}_x^T \boldsymbol{\alpha} \leq \lambda_i \ (\forall i : y_i = -1, \forall x \in B_i),$$

$$\sum_{z \in P_S} \sum_{v \in P_S} \alpha_z \alpha_v K(z, v) \leq 1.$$

It is well known that this is an SOCP problem. Moreover, it is clear that OP'_t for OP 2 can be formulated as an LP problem. We describe the algorithm for OP 1 in algorithm 2.

One might be concerned concern that a kernel matrix could become large when a sample consists a large number of bags and instances. However, note that the kernel matrix of $K(z, x)$, which is used in algorithm 2, needs to be computed only once at the beginning of algorithm 1, not at every iteration.

As a result, our learning algorithm outputs a classifier,

$$g(B) = \mathrm{sign}\left(\sum_{t=1}^{T} w_t \max_{x \in B} \sum_{z \in P_S} \alpha_{t,z} K(z, x) \right),$$

where w_t and $\boldsymbol{\alpha}_t$ are obtained in training phase. Therefore, the computational cost for predicting the label of B is $O(T|P_S||B|)$ in the worst case when all elements of $\alpha_{t,z}$ are nonzero. However, when we employ our sparse formulation OP 2, which allows us to find a sparse α, the computational cost is expected to be much smaller than the worst case.

5 Generalization Bound of the Hypothesis Class

In this section, we provide a generalization bound of hypothesis classes $\mathrm{conv}(H_U)$ for various U and K.

Let $\Phi(P_S) = \{\Phi(z) \mid z \in P_S\}$. Let $\Phi_{\mathrm{diff}}(P_S) = \{\Phi(z) - \Phi(z') \mid z, z' \in P_S, z \neq z'\}$. By viewing each instance $\mathbf{v} \in \Phi_{\mathrm{diff}}(P_S)$ as a hyperplane $\{\mathbf{u} \mid \langle \mathbf{v}, \mathbf{u} \rangle = 0\}$, we can naturally define a partition of the Hilbert space $\mathbb{H}$ by the set of all hyperplanes $\mathbf{v} \in \Phi_{\mathrm{diff}}(P_S)$. Let $\mathcal{I}$ be the set of all cells of the partition, that is, $\mathcal{I} = \{I \mid I = \cap_{\mathbf{v} \in V} \{\mathbf{u} \mid \langle \mathbf{v}, \mathbf{u} \rangle > 0\}, I \neq \emptyset, V \subseteq \Phi_{\mathrm{diff}}(P_S), \mathbf{v} \in V \Leftrightarrow -\mathbf{v} \notin V \text{ for all } \mathbf{v} \in \Phi_{\mathrm{diff}}(P_S)\}$. Each cell $I \in \mathcal{I}$ is a polyhedron defined by a minimal set $V_I \subseteq \Phi_{\mathrm{diff}}(P_S)$ that satisfies $I = \bigcap_{\mathbf{v} \in V_I} \{\mathbf{u} \mid \langle \mathbf{u}, \mathbf{v} \rangle > 0\}$. Let

$$\mu^* = \min_{I \in \mathcal{I}} \max_{\mathbf{u} \in I \cap U} \min_{\mathbf{v} \in V_I} |\langle \mathbf{u}, \mathbf{v} \rangle|.$$

Let $d^*_{\Phi,S}$ be the VC dimension of the set of linear classifiers over the finite set $\Phi_{\mathrm{diff}}(P_S)$, given by $F_U = \{f : \mathbf{v} \mapsto \mathrm{sign}(\langle \mathbf{u}, \mathbf{v} \rangle) \mid \mathbf{u} \in U\}$.

Then we have the following generalization bound on the hypothesis class of equation 1.2:

Theorem 2. *Let $\Phi : \mathcal{X} \to \mathbb{H}$. Suppose that for any $z \in \mathcal{X}$, $\|\Phi(z)\|_{\mathbb{H}} \leq R$. Then, for any $\rho > 0$, with high probability the following holds for any $g \in \mathrm{conv}(H_U)$ with $U \subseteq \{\mathbf{u} \in \mathbb{H} \mid \|\mathbf{u}\|_{\mathbb{H}} \leq 1\}$:*

$$\mathcal{E}_D(g) \leq \mathcal{E}_\rho(g) + O\left(\frac{R\sqrt{d^*_{\Phi,S} \log |P_S|}}{\rho\sqrt{m}} \right), \tag{5.2}$$

where (i) for any Φ, $d^*_{\Phi,S} = O((R/\mu^*)^2)$, (ii) if $\mathcal{X} \subseteq \mathbb{R}^\ell$ and Φ is the identity mapping (i.e., the associated kernel is the linear kernel), or (iii) if $\mathcal{X} \subseteq \mathbb{R}^\ell$ and Φ satisfies the condition that $\langle \Phi(z), \Phi(x) \rangle$ is monotone decreasing with respect to

$\|z-x\|_2$ *(e.g., the mapping defined by the gaussian kernel) and* $U=\{\Phi(z)\mid z\in\mathbb{R}^\ell, \|\Phi(z)\|_{\mathbb{H}}\le 1\}$, *then* $d^*_{\Phi,S}=O(\min((R/\mu^*)^2,\ell))$.

We show the proof in appendix B.

5.1 Comparison with the Existing Bounds. A similar generalization bound can be derived from a known bound of the Rademacher complexity of H_U (see theorem 20 of Sabato & Tishby, 2012) and a generalization bound of $\operatorname{conv}(H)$ for any hypothesis class H (see corollary 6.1 of Mohri et al., 2012):

$$\mathcal{E}_D(g)\le\mathcal{E}_\rho(g)+O\left(\frac{\log\left(\sum_{i=1}^m|B_i|\right)\log(m)}{\rho\sqrt{m}}\right).$$

Note that Sabato and Tishby (2012) fixed $R=1$. For simplicity, we omit some constants of theorem 20 of Sabato and Tishby (2012). Note that $|P_S|\le\sum_{i=1}^m|B_i|$ by definition. The bound above is incomparable to theorem 2 in general, as ours uses the parameter $d^*_{\Phi,S}$ and the other has the extra $\sqrt{\log\left(\sum_{i=1}^m|B_i|\right)}\log(m)$ term. However, our bound is better in terms of the sample size m by the factor of $O(\log m)$ when other parameters are regarded as constants.

6 SL by MIL

6.1 Time-Series Classification with Shapelets. In the following, we introduce a framework of time-series classification problem based on shapelets (i.e., the SL problem). As mentioned in the previous section, a time-series $\boldsymbol{\tau}=(\tau[1],\dots,\tau[L])\in\mathbb{R}^L$ can be identified with a bag $B_{\boldsymbol{\tau}}=\{(\tau[j],\dots,\tau[j+\ell-1])\mid 1\le j\le L-\ell+1\}$ that consists of all subsequences of $\boldsymbol{\tau}$ of length ℓ. The learner receives a labeled sample $S=((B_{\boldsymbol{\tau}_1},y_1),\dots,(B_{\boldsymbol{\tau}_m},y_m))\in(2^{\mathbb{R}^\ell}\times\{-1,1\})^m$, where each labeled bag (i.e., labeled time series) is independently drawn according to some unknown distribution D over a finite support of $2^{\mathbb{R}^\ell}\times\{-1,+1\}$. The goal of the learner is to predict the labels of an unseen time series correctly. In this way, the SL problem can be viewed as an MIL problem, and thus we can apply our algorithms and theory.

Note that for time-series classification, various similarity measures can be represented by a kernel—for example, the gaussian kernel (behaves like the Euclidean distance) and the dynamic time warping (DTW) kernel. Moreover, our framework can generally apply to non-real-valued sequence data (e.g., text, and a discrete signal) using a string kernel.

6.2 Our Theory and Algorithms for SL. By theorem 2, we can immediately obtain the generalization bound of our hypothesis class in SL as follows:

Corollary 1. *Consider time-series sample S of size m and length L. For any fixed $\ell < L$, the following generalization error bound holds for all $g \in \mathrm{conv}(H_U)$ in which the length of shapelet is ℓ:*

$$\mathcal{E}_D(g) \leq \mathcal{E}_\rho(g) + O\left(\frac{R\sqrt{d^*_{\Phi,S}\log(m(L-\ell+1))}}{\rho\sqrt{m}}\right).$$

To the best of our knowledge, this is the first result on the generalization performance of SL.

Theorem 1 gives justification to the heuristics that choose the shapelets extracted from the instances appearing in the training sample (i.e., the sub-sequences for SL tasks). Moreover, several methods using a linear combination of shapelet-based classifiers (e.g., Hills et al., 2014; Grabocka et al., 2014) are supported by corollary 1.

For time-series classification problems, shapelet-based classification has a greater advantage of the interpretability or visibility than other time-series classification methods (see, e.g., Ye & Keogh, 2009). Although we use a nonlinear kernel function, we can observe important sub-sequences that contribute to effective shapelets by solving OP 2 because of the sparsity (see also the experimental results). Moreover, for unseen time-series data, we can observe the types of sub-sequences that contribute to the predicted class by observing maximizer $x \in B$.

6.3 Learning Shapelets of Different Lengths.

For time-series classification, many existing methods take advantage of using shapelets of various lengths. Below, we show that our formulation can be easily applied to the case.

A time series $\boldsymbol{\tau} = (\tau[1], \ldots, \tau[L]) \in \mathbb{R}^L$ can be identified with a bag $B_{\boldsymbol{\tau}} = \{(\tau[j], \ldots, \tau[j+\ell-1]) \mid 1 \leq j \leq L-\ell+1, \forall \ell \in Q\}$ that consists of all length $\ell \in Q \subseteq \{1, \ldots, L\}$ of sub-sequences of $\boldsymbol{\tau}$. That is, this is also a special case of MIL that a bag contains different dimensional instances.

There is a simple way to apply our learning algorithm to this case. We employ some kernels $K(z, x)$ that support different dimensional instance pairs z and x. Fortunately, such kernels have been studied well in the time-series domain. For example, the DTW kernel and global alignment kernel (Cuturi, 2011) are well-known kernels that support time series of different lengths. However, the size of the kernel matrix of $K(z, x)$ becomes $m(\sum_{\ell \in Q}(L-\ell+1))^2$. In practice, it requires a high memory cost for large time-series data. Moreover, in general, the above kernel requires a higher computational cost than standard kernels.

We introduce a practical way to learn shapelets of different lengths based on heuristics. In each weak learning problem, we decomposed the original weak learning problem over different dimensional data space into the weak learning problems over each dimensional data space. For example, we consider solving the following problem instead of the weak learning problem OP 1,

$$\begin{aligned} &\min_{\ell}\min_{\boldsymbol{\alpha}} \quad -\sum_{i=1}^{m} d_i y_i \max_{x\in B_i^{\ell}} \sum_{z\in {P_S}^{\ell}} \alpha_z K(z,x), \\ &\text{sub.to} \quad \sum_{z\in {P_S}^{\ell}} \sum_{v\in {P_S}^{\ell}} \alpha_z \alpha_v K(z,v) \le 1, \end{aligned}$$

where B_i^{ℓ} denotes the ℓ dimensional instances (i.e., length ℓ of sub-sequences) in B_i, and ${P_S}^{\ell}$ denotes $\bigcup_{i=1}^{m} B_i^{\ell}$. The total size of kernel matrices becomes $m\sum_{\ell\in Q}((L-\ell+1))^2$, and thus this method does not require such a large kernel matrix. Moreover, in this way, we do not need to use a kernel that supports different dimensional instances. Note that even using this heuristic, the obtained final hypothesis has theoretical generalization performance. This is because the hypothesis class is still represented as the form of equation 2.1. In our experiment, we use the latter method by giving weight to memory efficiency.

6.4 Heuristics for Computational Efficiency. For the practical applications, we introduce some heuristics for improving efficiency in our algorithm.

6.4.1 Reduction of P_S. Especially for time-series data, the size $|P_S|$ often becomes large because $|P_S| = O(mL)$. Therefore, constructing a kernel matrix of $|P_S| \times |P_S|$ has high computational costs for time-series data. For example, when we consider sub-sequences as instances for time-series classification, we have a large computational cost because of the number of sub-sequences of training data (e.g., approximately 10^6 when the sample size is 1000 and the length of each time series is 1000, which results in a similarity matrix of size 10^{12}). However, in most cases, many sub-sequences in time-series data are similar to each other. Therefore, we only use representative instances $\hat{P}_S$ instead of the set of all instances P_S. In this letter, we use k-means clustering to reduce the size of $|P_S|$. Note that our heuristic approach is still supported by our theoretical generalization error bound. This is because the hypothesis set $H_{U'}$ with the reduced shapelets U' is the subset of H_U, and the Rademacher complexity of $H_{U'}$ is exactly smaller than the Rademacher complexity of H_U. Thus, theorem 2 holds for the hypothesis class considering the set H_U of all possible shapelets U, and thus that theorem also holds for the hypothesis class using the set $H_{U'}$ of some reduced shapelets U'. Although this approach may decrease the training classification accuracy in practice, it drastically decreases the computational cost for a large data set.

6.4.2 Initialization in Weak Learning Problem. The DC program may slowly converge to a local optimum depending on the initial solution. In algorithm 2, we fix an initial $\boldsymbol{\alpha}_0$ as follows. More precisely, we initially solve

$$\begin{aligned}\boldsymbol{\alpha}_0 = \arg\max_{\boldsymbol{\alpha}} \ & \sum_{i=1}^{m} d_i y_i \max_{x \in B_i} \sum_{z \in P_S} \alpha_z K(z, x), \\ \text{sub.to} \ & \boldsymbol{\alpha} \text{ is a one-hot vector.}\end{aligned} \tag{6.1}$$

That is, we choose the most discriminative shapelet from P_S as the initial point of $\mathbf{u}$ for given $\mathbf{d}$. We expect that it will speed up the convergence of the loop of line 3, and the obtained classifier is better than the methods that choose effective shapelets from subsequences.

7 Experiments

In this section, we show some experimental results implying that our algorithm performs comparably to the existing shapelet-based classifiers for both SL and MIL tasks.

7.1 Results for Time-Series Data.

We use several binary labeled data sets[2] in UCR data sets (Chen et al., 2015), which are often used as benchmark data sets for time-series classification methods. We used a weak learning problem OP 2 because the interpretability of the obtained classifier is required in shapelet-based time-series classification.

We compare the following three shapelet-based approaches:

- Shapelet transform (ST) provided by Bagnall, Lines, Bostrom, Large, and Keogh (2017)
- Learning time-series shapelets (LTS) provided by Grabocka et al. (2014)
- Our algorithm using shapelets of different lengths (which we will refer to as Ours)

We used the implementation of ST provided by Löning et al. (2019), and used the implementation of LTS provided by Tavenard, Faouzi, and Vandewiele (2017). The classification rule of shapelets transform has the form

$$g(B) = f\left(\max_{x \in B} -\|z_1 - x\|, \ldots, \max_{x \in B} -\|z_k - x\|\right),$$

where f is a user-defined classification function (the implementation employs decision forest), $z_1, \ldots, z_k \in P_S$ (in the time-series domain, this z_j is

[2] Note that our method is applicable to multiclass classification tasks by easy expansion (e.g., Platt, Cristianini, & Shawe-Taylor, 2000).

called a shapelet). The shapelets are chosen from training sub-sequences in some complicated way before learning f. The classification rule of learning time-series shapelets has the form

$$g(B) = \sum_{j=1}^{k} w_j \max_{x \in B} -\|z_j - x\|,$$

where $w_j \in \mathbb{R}$ and $z_j \in \mathbb{R}^\ell$ are learned parameters and the number of desired shapelets k is a hyperparameter.

Below we show the detailed condition of the experiment. For ST, we set the shapelet lengths $\{2, \ldots, L/2\}$, where L is the length of each time series in the data set. ST also requires a parameter of a time limit for searching shapelets, and we set it as 5 hours for each data set. For LTS, we used the hyperparameter sets (e.g., regularization parameter, number of shapelets) that the authors recommended in their website,[3] and we found an optimal hyperparameter by 3-fold cross-validation for each data set. For our algorithms, we implemented a weak learning algorithm that supports shapelets of different lengths (see section 6.3). In this experiment, we consider the case that each bag contains lengths $\{0.05, 0.1, 0.15, \ldots, 0.5\} \times L$ of the sub-sequences. We used the gaussian kernel $K(x, x') = \exp(-\frac{\|x-x'\|^2}{\ell\sigma^2})$ and chose $1/\sigma^2$ from $\{0.01, 0.05, 0.1, \ldots, 50\}$. We chose ν from $\{0.1, 0.2, 0.3, 0.4\}$. We use 100-means clustering with respect to each class to reduce P_S. The parameters we should tune are only ν and σ. We tuned these parameters via a procedure we give in appendix B.1. As an LP solver for WeakLearn and LPBoost, we used the CPLEX software. In addition to Ours, LTS employs k-means clustering to set the initial shapelets in the optimization algorithm. Therefore, we show the average accuracies for LTS and Ours considering the randomness of k-means clustering.

The classification accuracy results are shown in Table 1. We can see that our algorithms achieve performance comparable to that of ST and LTS. We conducted the Wilcoxon signed-rank test between Ours and the others. The p-value of the Wilcoxon signed-rank test for Ours and ST is 0.1247. The p-value of the Wilcoxon signed-rank test for Ours and LTS is 0.6219. The p-values are higher than 0.05, and thus we cannot rejcect that there is no significant difference between the medians of the accuracies. We can say that our MIL algorithm works well for time-series classification tasks without using domain-specific knowledge.

To compare the computation time of these methods, we selected the data sets for which these three methods have achieved similar performance. The experiments were performed on Intel Xeon Gold 6154, 36 core CPU, and 192 GB memory. Table 2 compares the running times of the training. Note

[3] http://fs.ismll.de/publicspace/LearningShapelets/.

Table 1: Classification Accuracies for Time-Series Data Sets.

Data Set	ST	LTS	Ours
BeetleFly	0.8	0.765	**0.835**
BirdChicken	0.9	0.93	**0.935**
Coffee	0.964	**1**	0.964
Computers	**0.704**	0.619	0.623
DistalPhalanxOutlineCorrect	0.757	0.714	**0.802**
Earthquakes	0.741	**0.748**	0.728
ECG200	0.85	0.835	**0.872**
ECGFiveDays	0.999	0.961	**1**
FordA	0.856	**0.914**	0.89
FordB	0.74	**0.9**	0.786
GunPoint	**0.987**	0.971	**0.987**
Ham	0.762	**0.782**	0.698
HandOutlines	**0.919**	0.892	0.87
Herring	0.594	**0.652**	0.588
ItalyPowerDemand	0.947	**0.951**	0.943
Lightning2	0.639	0.695	**0.779**
MiddlePhalanxOutlineCorrect	**0.794**	0.579	0.632
MoteStrain	**0.927**	0.849	0.845
PhalangesOutlinesCorrect	0.773	0.633	**0.792**
ProximalPhalanxOutlineCorrect	**0.869**	0.742	0.844
ShapeletSim	0.994	0.989	**1**
SonyAIBORobotSurface1	**0.932**	0.903	0.841
SonyAIBORobotSurface2	**0.922**	0.895	0.887
Strawberry	0.941	0.844	**0.947**
ToeSegmentation1	**0.956**	0.947	0.906
ToeSegmentation2	0.792	**0.886**	0.823
TwoLeadECG	**0.995**	0.981	0.949
Wafer	**1**	0.993	0.991
Wine	**0.741**	0.487	0.72
WormsTwoClass	**0.831**	0.752	0.608
Yoga	**0.847**	0.69	0.804

Note: The best accuracies are highlighted in bold.

Table 2: Training Time (Sec.) for Several Time-Series Data Sets.

Data Set	Number of Training Data	Length	ST	LTS	Ours
Earthquakes	322	512	18,889.8	250.5	1339.2
GunPont	50	150	18,016.2	22.3	36.9
ItalyPowerDemand	67	24	18,000.8	11.5	8.6
ShapeletSim	20	180	18,011.6	30.4	32.8
Wafer	1000	152	18,900.8	91.5	431.7

Table 3: Testing Time (Sec.) for Several Time-Series Data Sets.

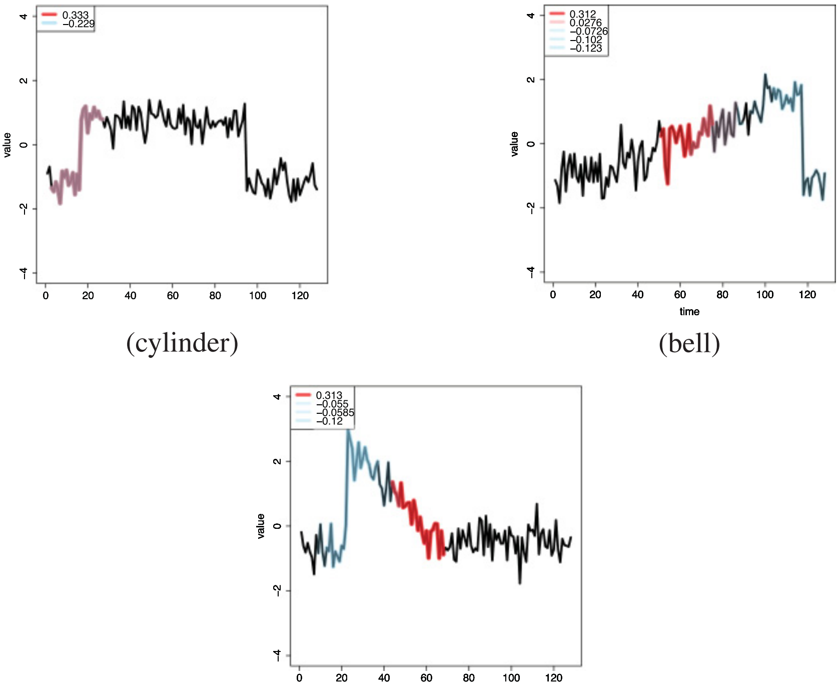
Data Set	Number of Test Data	Length	ST	LTS	Ours
Earthquakes	139	512	389.7	2.75	11.55
GunPont	150	150	48.0	1.1	3.9
ItalyPowerDemand	1029	24	3.3	0.5	10.7
ShapeletSim	180	180	104.0	1.8	1.1
Wafer	6164	152	5688.2	4.3	173.1

that again, for ST, we set the limitation of the running time as 5 hours for finding good shapelets. This running time limitation is a hyperparameter of the code, and it is difficult to estimat it before experiments. LTS efficiently worked compared with ST and Ours. However, it seems that LTS achieved lower performance than ST and Ours on accuracy. Table 3 shows the testing time of the methods. LTS also efficiently worked, simply because it finds effective shapelets of a fixed number (hyperparameter). ST and Ours may find a large number of shapelets, and this increases the computation time of prediction. For the Wafer data set, ST and Ours required a large computation time compared with LTS.

We cannot fairly compare the efficiency of these methods because the implementation environments (e.g., programming languages) are different. However, we can say that the proposed method achieved high classification accuracy with reasonable running time for training and prediction.

7.1.1 Interpretability of Our Method. We would like to show the interpretability of our method. We use the CBF data set, which contains three classes (cylinder, bell, and funnel) of time series. The reason is that it is known that the discriminative patterns are clear, and thus we can easily ascertain if the obtained hypothesis can capture the effective shapelets. For simplicity, we obtain a binary classification model for each class preparing one-versus-others training set. We used Ours with fixed shapelet length $\ell = 25$. We now introduce two types of visualization approaches to interpret a learned model.

One is the visualization of the characteristic sub-sequences of an input time series. When we predict the label of the time series B, we calculate a maximizer x^* in B for each $h_{\mathbf{u}}$, that is, $x^* = \arg\max_{x \in B} \langle \mathbf{u}, \Phi(x) \rangle$. For image recognition tasks, the maximizers are commonly used to observe the subimages that characterize the class of the input image (e.g., Chen et al., 2006). In time-series classification tasks, the maximizers also can be used to observe some characteristic sub-sequences. Figure 1 is an example of a visualization of maximizers. Each value in the legend indicates $w_{\mathbf{u}} \max_{x \in B} \langle \mathbf{u}, \Phi(x) \rangle$. That is, sub-sequences with positive values contribute to the positive class, and sub-sequences with negative values contribute to the negative class. Such



(cylinder)

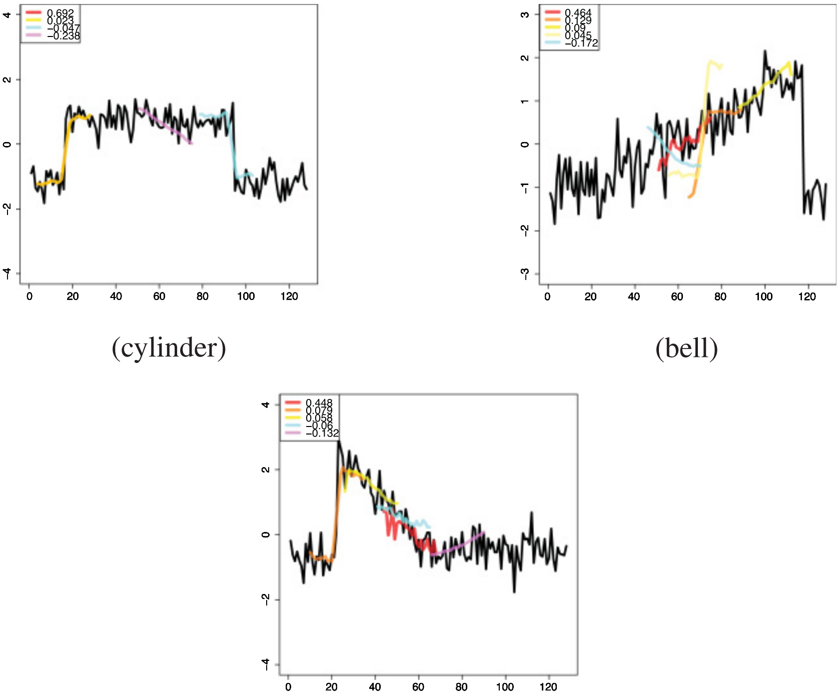
(bell)

(funnel)

Figure 1: Examples of the visualization of maximizers for CBF time-series data. Black lines are the original time series. We highlight each sub-sequence that maximizes the similarity with some shapelet in a classifier. Sub-sequences with positive values (red) contribute to the positive class, and sub-sequences with negative values (blue) contribute to the negative class.

visualization provides the sub-sequences that characterize the class of the input time series. For the cylinder class, although both positive and negative patterns match almost the same sub-sequence, the positive pattern is stronger than the negative, and thus the hypothesis can correctly discriminate the time series. For the bell and funnel classes, we can observe that the highlighted sub-sequences clearly indicate the discriminative patterns.

The other is the visualization of a final hypothesis $g(B) = \sum_{j=1}^{t} w_j h_j(B)$, where $h_j(B) = \max_{x \in B} \sum_{z_j \in \hat{P}_S} \alpha_{j,z_j} K(z_j, x)$ ($\hat{P}_S$ is the set of representative sub-sequences obtained by k-means clustering). Figure 2 is an example of the visualization of a final hypothesis obtained by our algorithm. The colored lines are all the z_js in g where both w_j and α_{j,z_j} were nonzero. Each legend value shows the multiplication of w_j and α_{j,z_j} corresponding to z_j. That is, positive values of the colored lines indicate the contribution rate for the



(cylinder)

(bell)

(funnel)

Figure 2: Examples of the visualization of shapelets for CBF time-series data. The colored lines show important patterns of the obtained classifier. Positive values on the colored lines (red to yellow) indicate the contribution rate for the positive class, and negative values (blue to purple) indicate the contribution rate for the negative class.

positive class, and negative values indicate the contribution rate for the negative class. Note that because it is difficult to visualize the shapelets over the Hilbert space associated with the gaussian kernel, we plotted each of them to match the original time series based on the Euclidean distance. Unlike the previous visualization analyses (see, e.g., Ye & Keogh, 2009), our visualization does not exactly interpret the final hypothesis because of the nonlinear feature map. However, we can deduce that the colored lines represent "important patterns," which make significant contributions to classification.

7.2 Results for Multiple-Instance Data. We selected the baselines of MIL algorithms as mi-SVM and MI-SVM (Andrews et al., 2003) and MILES (Chen, Bi, & Wang, 2006). mi-SVM and MI-SVM are classic methods in

Table 4: Details of MIL Data Sets.

Data Set	Sample Size	Number of Instances	Dimension
MUSK1	92	476	166
MUSK2	102	6598	166
elephant	200	1391	230
fox	200	1320	230
tiger	200	1220	230

MIL that still perform favorably compared with state-of-the-art methods for standard multiple-instance data (see, e.g., Doran, 2015). The details of the data sets are shown in Table 4.

mi- and MI-SVM find a single but optimized shapelet $\mathbf{u}$, which is not limited to the instance in the training sample. The classifiers obtained by these algorithms are formulated as

$$g(B) = \max_{x \in B} \langle \mathbf{u}, \Phi(x) \rangle = \max_{x \in B} \sum_{z \in P_S} \alpha_z K(z, x). \tag{7.1}$$

MILES finds the multiple-shapelets, but they are limited to the instances in the training sample. The classifier of MILES is formulated as follows:

$$g(B) = \sum_{z \in P_S} w_z \max_{x \in B} K(z, x). \tag{7.2}$$

We used the implementation provided by Doran[4] for mi-SVM and MI-SVM. We combined the gaussian kernel with mi-SVM and MI-SVM. Parameter C was chosen from $\{1, 10, 100, 1000, 10{,}000\}$. For our method and MILES,[5] we chose ν from $\{0.5, 0.3, 0.2, 0.15, 0.1\}$, and we used only the gaussian kernel. Furthermore, we chose σ from $\{0.005, 0.01, 0.05, 0.1, 0.5, 1.0\}$. We use 100-means clustering with respect to each class to reduce P_S. To avoid the randomness of k-means, we ran training 30 times and selected the model that achieved the best training accuracy. For efficiency, we demonstrated the weak learning problem OP 2. For all these algorithms, we estimated an optimal parameter set via 5-fold cross-validation. We used well-known multiple-instance data, as shown on the left-hand side of Table 5. The accuracies resulted from 10 times of 5-fold cross-validation.

The results are shown in Table 5. MILES and Ours achieve significantly better performance than mi- and MI-SVM. Ours achieves comparable performance to MILES. Table 6 shows the training accuracies of MILES and

[4] https://github.com/garydoranjr/misvm.

[5] MILES uses 1-norm SVM to obtain a final classifier. We implemented 1-norm SVM by using the formulation of Warmuth, Glocer, and Rätsch (2008).

Table 5: Classification Accuracies for MIL Data Sets.

Data Set	mi-SVM	MI-SVM	MILES	Ours
MUSK1	0.834 ± 0.084	0.820 ± 0.081	**0.865 ± 0.068**	0.844 ± 0.076
MUSK2	0.749 ± 0.082	0.840 ± 0.074	0.871 ± 0.072	**0.879 ± 0.067**
elephant	0.785 ± 0.070	0.823 ± 0.056	0.796 ± 0.068	**0.828 ± 0.061**
fox	0.618 ± 0.069	0.578 ± 0.075	**0.675 ± 0.071**	0.646 ± 0.063
tiger	0.752 ± 0.078	0.815 ± 0.055	**0.827 ± 0.057**	0.817 ± 0.058

Note: The best accuracies are highlighted in bold.

Table 6: Training Accuracies for MIL Data Sets.

Data Set	MILES	Ours
MUSK1	0.987	0.985
MUSK2	0.980	0.993
elephant	0.963	0.993
fox	0.987	0.995
tiger	0.973	0.993

Table 7: Training Time (Sec.) for MIL Data Sets.

Data Set	mi-SVM	MI-SVM	MILES	Ours
MUSK1	29.6	28.1	0.584	5.57
MUSK2	3760.1	3530.0	103.1	80.5
elephant	240.6	130.3	5.84	8.30
fox	201.9	139.2	5.4	26.4
tiger	158.5	118.0	4.6	9.8

Ours. It can be seen that Ours achieves higher training accuracy. This result is theoretically reasonable because our hypothesis class is richer than that of MILES. However, this means that Ours has a higher overfitting risk than does MILES.

Table 7 shows the training time of the five methods. It is clear that MILES and Ours are more efficient than mi- and MI-SVM. The main reason is that mi- and MI-SVM solve quadratic programming (QP) problems, while MILES and Ours solve LP problems. On average, MILES worked more efficiently than Ours. However, for MUSK2, which has a large number of instances, Ours worked more efficiently than MILES.

The testing time of each algorithm is shown in Table 8. We can see that Ours is comparable to the other algorithms.

Table 8: Testing Time (Sec.) for MIL Data Sets.

Data Set	mi-SVM	MI-SVM	MILES	Ours
MUSK1	0.010	0.004	0.011	0.045
MUSK2	0.577	0.063	0.129	0.083
elephant	0.053	0.015	0.067	0.115
fox	0.078	0.025	0.118	0.145
tiger	0.059	0.012	0.065	0.118

8 Conclusion and Future Work

We proposed a new MIL formulation that provides a richer class of the final classifiers based on infinitely many shapelets. We derived the tractable formulation over infinitely many shapelets with theoretical support and provided an algorithm based on LPBoost and the DC (difference of convex) algorithm. Our result gives theoretical justification for some existing shapelet-based classifiers (e.g., Chen et al., 2006; Hills et al., 2014). The experimental results demonstrate that our approach uniformly works for SL and MIL tasks without introducing domain-specific parameters and heuristics and compares with the baselines of shapelet-based classifiers.

Especially for time-series classification, the number of instances usually becomes large. Although we took a heuristic approach in the experiment, we think it is not an essential solution to improve efficiency. We preliminarily implemented OP 1 with orthogonal random features (Yu et al., 2016) that can approximate the gaussian kernel accurately. It allows us to solve the primal problem of OP 1 directly and to avoid constructing a large kernel matrix. The implementation vastly improved the efficiency however, it did not achieve high accuracy as compared with solutions of OP 2 with the heuristics. For SL tasks, there are many successful efficient methods using some heuristics specialized in the time-series domain (Keogh & Rakthanmanon, 2013; Renard et al., 2015; Grabocka et al., 2015; Wistuba et al., 2015; Hou et al., 2016; Karlsson et al., 2016). We will explore many ways to improve efficiency for SL tasks.

We would also like to improve the generalization error bound. The generalization error bound that we provided in this letter is incomparable to the existing bound. We would like to show the tighter bound than the existing bound. Since we think it requires a more complex analysis, we reserve this for future work. Our heuristics might reduce the model complexity (i.e., the risk of overfitting); however, we do not know how the complexity can be reduced by our heuristics theoretically. To apply our method to various domains, we would like to explore the general techniques for reducing the overfitting risk of our method.

Appendix A: Proof of Theorem 1

First, we give a definition for convenience.

Definition 1 *(The set Θ of mappings from a bag to an instance). Given a sample $S = (B_1, \ldots, B_m)$. For any $\mathbf{u} \in U$, let $\theta_{\mathbf{u},\Phi} : \{B_1, \ldots, B_m\} \to \mathcal{X}$ be a mapping defined by*

$$\theta_{\mathbf{u},\Phi}(B_i) := \arg\max_{x \in B_i} \left\langle \mathbf{u}, \Phi\left(x\right)\right\rangle,$$

and we define the set of all $\theta_{\mathbf{u},\Phi}$ for S as $\Theta_{S,\Phi} = \{\theta_{\mathbf{u},\Phi} \mid \mathbf{u} \in U\}$. For the sake of brevity, $\theta_{\mathbf{u},\Phi}$ and $\Theta_{S,\Phi}$ will be abbreviated as $\theta_{\mathbf{u}}$ and Θ, respectively.

Following is the proof of theorem 1.

Proof. We can rewrite the optimization problem 3.3 by using $\theta \in \Theta$ as follows:

$$\begin{aligned} \max_{\theta \in \Theta} \max_{\mathbf{u} \in \mathbb{H}: \theta_{\mathbf{u}} = \theta} \quad & \sum_{i=1}^{m} y_i d_i \left\langle \mathbf{u}, \Phi\left(\theta(B_i)\right)\right\rangle \\ \text{sub.to} \quad & \|\mathbf{u}\|_{\mathbb{H}}^2 \leq 1. \end{aligned} \tag{A.1}$$

Thus, if we fix $\theta \in \Theta$, we have a subproblem. Since the constraint $\theta = \theta_{\mathbf{u}}$ can be written as the number $|P_S|$ of linear constraints (i.e., sub.to $\langle \mathbf{u}, \Phi(x)\rangle \leq \langle \mathbf{u}, \Phi(\theta(B_i))\rangle$ $(i \in [m], x \in B_i)$), each subproblem is equivalent to a convex optimization. Indeed, each subproblem can be written as the equivalent unconstrained minimization (by neglecting constants in the objective),

$$\min_{\mathbf{u} \in \mathbb{H}} \beta \|\mathbf{u}\|_{\mathbb{H}}^2 - \sum_{i=1}^{m} \sum_{x \in B_i} \eta_{i,x} \left(\left\langle \mathbf{u}, \Phi\left(\theta(B_i)\right)\right\rangle - \left\langle \mathbf{u}, \Phi(x)\right\rangle\right) - \sum_{i=1}^{m} y_i d_i \left\langle \mathbf{u}, \Phi\left(\theta(B_i)\right)\right\rangle,$$

where β and $\eta_{i,x}$ $(i \in [m], x \in B_i)$ are the corresponding positive constants. Now for each subproblem, we can apply the standard representer theorem argument (see, e.g., Mohri et al., 2012). Let $\mathbb{H}_1$ be the subspace $\{\mathbf{u} \in \mathbb{H} \mid \mathbf{u} = \sum_{z \in P_S} \alpha_z \Phi(z), \alpha_z \in \mathbb{R}\}$. We denote $\mathbf{u}_1$ as the orthogonal projection of $\mathbf{u}$ onto $\mathbb{H}_1$, and any $\mathbf{u} \in \mathbb{H}$ has the decomposition $\mathbf{u} = \mathbf{u}_1 + \mathbf{u}^{\perp}$. Since $\mathbf{u}^{\perp}$ is orthogonal with regard to $\mathbb{H}_1$, $\|\mathbf{u}\|_{\mathbb{H}}^2 = \|\mathbf{u}_1\|_{\mathbb{H}}^2 + \|\mathbf{u}^{\perp}\|_{\mathbb{H}}^2 \geq \|\mathbf{u}_1\|_{\mathbb{H}}^2$. On the other hand, $\left\langle \mathbf{u}, \Phi\left(z\right)\right\rangle = \left\langle \mathbf{u}_1, \Phi\left(z\right)\right\rangle$. Therefore, the optimal solution of each subproblem has to be contained in $\mathbb{H}_1$. This implies that the optimal solution, which is the maximum over all solutions of subproblems, is contained in $\mathbb{H}_1$ as well. □

Appendix B: Proof of Theorem 2

We use θ and Θ of definition 1.

Definition 2 *(The Rademacher and the Gaussian complexity; Bartlett & Mendelson, 2003). Given a sample $S = (x_1, \ldots, x_m) \in \mathcal{X}^m$, the empirical Rademacher complexity $\mathfrak{R}(H)$ of a class $H \subset \{h : \mathcal{X} \to \mathbb{R}\}$ with regard to S is defined as $\mathfrak{R}_S(H) = \frac{1}{m} \mathop{\mathrm{E}}_{\boldsymbol{\sigma}} \left[\sup_{h \in H} \sum_{i=1}^m \sigma_i h(x_i)\right]$, where $\boldsymbol{\sigma} \in \{-1, 1\}^m$, and each σ_i is an independent uniform random variable in $\{-1, 1\}$. The empirical gaussian complexity $\mathfrak{G}_S(H)$ of H with regard to S is defined similarly, but each σ_i is drawn independently from the standard normal distribution.*

The following bounds are well known:

Lemma 1 *(Lemma 4 of Bartlett & Mendelson, 2003). $\mathfrak{R}_S(H) = O(\mathfrak{G}_S(H))$.*

Lemma 2 *(Corollary 6.1 of Mohri et al., 2012). For fixed ρ, $\delta > 0$, the following bound holds with probability at least $1 - \delta$: for all $f \in \mathrm{conv}(H)$,*

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

To derive a generalization bound based on the Rademacher or the gaussian complexity is quite standard in the statistical learning theory literature and applicable to our classes of interest as well. However, a standard analysis provides us suboptimal bounds.

Lemma 3. *Suppose that for any $z \in \mathcal{X}$, $\|\Phi(z)\|_{\mathbb{H}} \leq R$. Then the empirical gaussian complexity of H_U with respect to S for $U \subseteq \{\mathbf{u} \mid \|\mathbf{u}\|_{\mathbb{H}} \leq 1\}$ is bounded as follows:*

$$\mathfrak{G}_S(H) \leq \frac{R\sqrt{(\sqrt{2} - 1) + 2(\ln |\Theta|)}}{\sqrt{m}}.$$

Proof. Since U can be partitioned into $\bigcup_{\theta \in \Theta} \{\mathbf{u} \in U \mid \theta_{\mathbf{u}} = \theta\}$,

$$\begin{aligned}
\mathfrak{G}_S(H_U) &= \frac{1}{m} \mathop{\mathrm{E}}_{\boldsymbol{\sigma}} \left[\sup_{\theta \in \Theta} \sup_{\mathbf{u} \in U : \theta_{\mathbf{u}} = \theta} \sum_{i=1}^m \sigma_i \left\langle \mathbf{u}, \Phi\left(\theta(B_i)\right) \right\rangle \right] \\
&= \frac{1}{m} \mathop{\mathrm{E}}_{\boldsymbol{\sigma}} \left[\sup_{\theta \in \Theta} \sup_{\mathbf{u} \in U : \theta_{\mathbf{u}} = \theta} \left\langle \mathbf{u}, \left(\sum_{i=1}^m \sigma_i \Phi\left(\theta(B_i)\right) \right) \right\rangle \right] \\
&\leq \frac{1}{m} \mathop{\mathrm{E}}_{\boldsymbol{\sigma}} \left[\sup_{\theta \in \Theta} \sup_{\mathbf{u} \in U} \left\langle \mathbf{u}, \left(\sum_{i=1}^m \sigma_i \Phi\left(\theta(B_i)\right) \right) \right\rangle \right] \\
&\leq \frac{1}{m} \mathop{\mathrm{E}}_{\boldsymbol{\sigma}} \left[\sup_{\theta \in \Theta} \left\| \sum_{i=1}^m \sigma_i \Phi\left(\theta(B_i)\right) \right\|_{\mathbb{H}} \right]
\end{aligned}$$

$$= \frac{1}{m} \mathop{\mathrm{E}}_{\boldsymbol{\sigma}} \left[\sup_{\theta \in \Theta} \sqrt{\left\| \sum_{i=1}^{m} \sigma_i \Phi\left(\theta(B_i)\right) \right\|_{\mathbb{H}}^2} \right]$$

$$= \frac{1}{m} \mathop{\mathrm{E}}_{\boldsymbol{\sigma}} \left[\sqrt{\sup_{\theta \in \Theta} \left\| \sum_{i=1}^{m} \sigma_i \Phi\left(\theta(B_i)\right) \right\|_{\mathbb{H}}^2} \right]$$

$$\leq \frac{1}{m} \sqrt{\mathop{\mathrm{E}}_{\boldsymbol{\sigma}} \left[\sup_{\theta \in \Theta} \left\| \sum_{i=1}^{m} \sigma_i \Phi\left(\theta(B_i)\right) \right\|_{\mathbb{H}}^2 \right]}. \tag{B.1}$$

The first inequality is derived from the relaxation of $\mathbf{u}$, the second inequality is due to Cauchy-Schwarz inequality and the fact $\|\mathbf{u}\|_{\mathbb{H}} \leq 1$, and the last inequality is due to Jensen's inequality. We denote by $\mathbf{K}^{(\theta)}$ the kernel matrix such that $\mathbf{K}_{ij}^{(\theta)} = \langle \Phi((\theta(B_i)), \Phi(\theta(B_j))\rangle$. Then we have

$$\mathop{\mathrm{E}}_{\boldsymbol{\sigma}} \left[\sup_{\theta \in \Theta} \left\| \sum_{i=1}^{m} \sigma_i \Phi\left(\theta(B_i)\right) \right\|_{\mathbb{H}}^2 \right] = \mathop{\mathrm{E}}_{\boldsymbol{\sigma}} \left[\sup_{\theta \in \Theta} \sum_{i,j=1}^{m} \sigma_i \sigma_j \mathbf{K}_{ij}^{(\theta)} \right]. \tag{B.2}$$

We now derive an upper bound of the right-hand side as follows.

For any $c > 0$,

$$\exp\left(c \mathop{\mathrm{E}}_{\boldsymbol{\sigma}} \left[\sup_{\theta \in \Theta} \sum_{i,j=1}^{m} \sigma_i \sigma_j \mathbf{K}_{ij}^{(\theta)} \right] \right)$$

$$\leq \mathop{\mathrm{E}}_{\boldsymbol{\sigma}} \left[\exp\left(c \sup_{\theta \in \Theta} \sum_{i,j=1}^{m} \sigma_i \sigma_j \mathbf{K}_{ij}^{(\theta)} \right) \right]$$

$$= \mathop{\mathrm{E}}_{\boldsymbol{\sigma}} \left[\sup_{\theta \in \Theta} \exp\left(c \sum_{i,j=1}^{m} \sigma_i \sigma_j \mathbf{K}_{ij}^{(\theta)} \right) \right]$$

$$\leq \sum_{\theta \in \Theta} \mathop{\mathrm{E}}_{\boldsymbol{\sigma}} \left[\exp\left(c \sum_{i,j=1}^{m} \sigma_i \sigma_j \mathbf{K}_{ij}^{(\theta)} \right) \right].$$

The first inequality is due to Jensen's inequality, and the second inequality is due to the fact that the supremum is bounded by the sum. By using the symmetry property of $\mathbf{K}^{(\theta)}$, we have $\sum_{i,j=1}^{m} \sigma_i \sigma_j \mathbf{K}_{ij}^{(\theta)} = \boldsymbol{\sigma}^\top \mathbf{K}^{(\theta)} \boldsymbol{\sigma}$, which is rewritten as

$$\top\boldsymbol{\sigma}\mathbf{K}^{(\theta)}\boldsymbol{\sigma} = \top(\top\mathbf{V}\boldsymbol{\sigma})\begin{pmatrix}\lambda_1^{(\theta)} & & 0\\ & \ddots & \\ 0 & & \lambda_m^{(\theta)}\end{pmatrix}\top\mathbf{V}\boldsymbol{\sigma},$$

where $\lambda_1^{(\theta)} \geq \cdots \geq \lambda_m^{(\theta)} \geq 0$ are the eigenvalues of $\mathbf{K}^{(\theta)}$ and $\mathbf{V} = (\mathbf{v}_1, \ldots, \mathbf{v}_m)$ is the orthonormal matrix such that $\mathbf{v}_i$ is the eigenvector that corresponds to the eigenvalue λ_i. By the reproductive property of gaussian distribution, $\top\mathbf{V}\boldsymbol{\sigma}$ obeys the same gaussian distribution as well, so

$$\begin{aligned}
&\sum_{\theta\in\Theta}\mathop{\mathrm{E}}_{\boldsymbol{\sigma}}\left[\exp\left(c\sum_{i,j=1}^{m}\sigma_i\sigma_j\mathbf{K}_{ij}^{(\theta)}\right)\right]\\
&= \sum_{\theta\in\Theta}\mathop{\mathrm{E}}_{\boldsymbol{\sigma}}\left[\exp\left(c\top\boldsymbol{\sigma}\mathbf{K}^{(\theta)}\boldsymbol{\sigma}\right)\right]\\
&= \sum_{\theta\in\Theta}\mathop{\mathrm{E}}_{\boldsymbol{\sigma}}\left[\exp\left(c\sum_{k=1}^{m}\lambda_k^{(\theta)}(\top\mathbf{v}_k\boldsymbol{\sigma})^2\right)\right]\\
&= \sum_{\theta\in\Theta}\Pi_{k=1}^{m}\mathop{\mathrm{E}}_{\sigma_k}\left[\exp\left(c\lambda_k^{(\theta)}\sigma_k^2\right)\right] \quad (\text{replace } \boldsymbol{\sigma} = \top\mathbf{v}_k\boldsymbol{\sigma})\\
&= \sum_{\theta\in\Theta}\Pi_{k=1}^{m}\left(\int_{-\infty}^{\infty}\exp\left(c\lambda_k^{(\theta)}\sigma^2\right)\frac{\exp(-\sigma^2)}{\sqrt{2\pi}}d\sigma\right)\\
&= \sum_{\theta\in\Theta}\Pi_{k=1}^{m}\left(\int_{-\infty}^{\infty}\frac{\exp(-(1-c\lambda_k^{(\theta)})\sigma^2)}{\sqrt{2\pi}}d\sigma\right).
\end{aligned}$$

Now we replace σ by $\sigma' = \sqrt{1-c\lambda_k^{(\theta)}}\sigma$. Since $d\sigma' = \sqrt{1-c\lambda_k^{(\theta)}}d\sigma$, we have:

$$\begin{aligned}
&\int_{-\infty}^{\infty}\frac{\exp(-(1-c\lambda_k^{(\theta)})\sigma^2)}{\sqrt{2}\pi}d\sigma = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty}\frac{\exp(-\sigma'^2)}{\sqrt{1-c\lambda_k^{(\theta)}}}d\sigma'\\
&= \frac{1}{\sqrt{1-c\lambda_k^{(\theta)}}}.
\end{aligned}$$

Now, applying the inequality that $\frac{1}{\sqrt{1-x}} \leq 1 + 2(\sqrt{2}-1)x$ for $0 \leq x \leq \frac{1}{2}$, the bound becomes

$$\exp\left(c\mathop{\mathrm{E}}_{\boldsymbol{\sigma}}\left[\sup_{\theta\in\Theta}\sum_{i,j=1}^{m}\sigma_i\sigma_j\mathbf{K}_{ij}^{(\theta)}\right]\right)$$

$$\leq \sum_{\theta\in\Theta} \Pi_{k=1}^{m} \left(1 + 2(\sqrt{2}-1)c\lambda_k^{(\theta)} + 2\lambda_1\right). \tag{B.3}$$

Further, taking the logarithm, dividing the both sides by c, letting $c = \frac{1}{2\max_k \lambda_k^{(\theta)}} = 1/(2\lambda_1^{(\theta)})$, fix $\theta = \theta^*$ such that θ^* maximizes equation B.3, and applying $\ln(1+x) \leq x$, we get:

$$\begin{aligned}
&\operatorname*{E}_{\boldsymbol{\sigma}}\left[\sup_{\theta\in\Theta}\sum_{i,j=1}^{m}\sigma_i\sigma_j\mathbf{K}_{ij}^{(\theta^*)}\right]\\
&\leq (\sqrt{2}-1)\sum_{k=1}^{m}\lambda_k^{(\theta^*)} + 2\lambda_1^{(\theta^*)}\ln|\Theta|\\
&= (\sqrt{2}-1)\top(\mathbf{K}^{(\theta^*)}) + 2\lambda_1^{(\theta^*)}\ln|\Theta|\\
&\leq (\sqrt{2}-1)mR^2 + 2mR^2\ln|\Theta|,
\end{aligned} \tag{B.4}$$

where the last inequality holds since $\lambda_1^{(\theta^*)} = \|\mathbf{K}^{(\theta^*)}\|_2 \leq m\|\mathbf{K}^{(\theta)}\|_{\max} \leq R^2$. By equations B.1 and B.4, we have

$$\begin{aligned}
\mathfrak{G}_S(H) &\leq \frac{1}{m}\sqrt{\operatorname*{E}_{\boldsymbol{\sigma}}\left[\sup_{\theta\in\Theta}\sum_{i,j=1}^{m}\sigma_i\sigma_j\mathbf{K}_{ij}^{(\theta)}\right]}\\
&\leq \frac{R\sqrt{(\sqrt{2}-1)+2\ln|\Theta|}}{\sqrt{m}}.
\end{aligned}$$

□

Thus, it suffices to bound the size $|\Theta|$. The basic idea to get our bound is the following geometric analysis. Fix any $i \in [m]$ and consider points $\{\Phi(x) \mid x \in B_i\}$. Then we define equivalence classes of $\mathbf{u}$ such that $\theta_{\mathbf{u}}(i)$ is in the same class, which defines a Voronoi diagram for the points $\{\Phi(x) \mid x \in B_i\}$. Note here that the similarity is measured by the inner product, not a distance. More precisely, let $\{V_i(x) \mid x \in B_i\}$ be the Voronoi diagram, with each region defined as $V_i(x) = \{\mathbf{u} \in \mathbb{H} \mid \theta_{\mathbf{u}}(B_i) = x\}$ Let us consider the set of intersections $\bigcap_{i\in[m]} V_i(x_i)$ for all combinations of $(x_1, \ldots, x_m) \in B_1 \times \cdots \times B_m$. The key observation is that each nonempty intersection corresponds to a mapping $\theta_{\mathbf{u}} \in \Theta$. Thus, we obtain $|\Theta| =$ (the number of intersections $\bigcap_{i\in[m]} V_i(x_i)$). In other words, the size of Θ is exactly the number of rooms defined by the intersections of m Voronoi diagrams $V_1, \ldots, V_m$. From now on, we will derive the upper bound based on this observation.

Lemma 4.

$$|\Theta| = O(|P_S|^{2d^*_{\Phi,S}}).$$

Proof. We will reduce the problem of counting intersections of the Voronoi diagrams to that of counting possible labelings by hyperplanes for some set. Note that for each neighboring Voronoi region, the border is a part of hyperplane since the closeness is defined in terms of the inner product. Therefore, by simply extending each border to a hyperplane, we obtain intersections of half-spaces defined by the extended hyperplanes. Note that the size of these intersections gives an upper bound of intersections of the Voronoi diagrams. More precisely, we draw hyperplanes for each pair of points in $\Phi(P_S)$ so that each point on the hyperplane has the same inner product between two points. Note that for each pair $\Phi(z)$, $\Phi(z') \in P_S$, the normal vector of the hyperplane is given as $\Phi(z) - \Phi(z')$ (by fixing the sign arbitrary). Thus, the set of hyperplanes obtained by this procedure is exactly $\Phi_{\text{diff}}(P_S)$. The size of $\Phi_{\text{diff}}(P_S)$ is $\binom{|P_S|}{2}$, which is at most $|P_S|^2$. Now, we consider a dual space by viewing each hyperplane as a point and each point in U as a hyperplane. Points $\mathbf{u}$ (hyperplanes in the dual) in an intersection give the same labeling on the points in the dual domain. Therefore, the number of intersections in the original domain is the same as the number of the possible labelings on $\Phi_{\text{diff}}(P_S)$ by hyperplanes in U. By the classical Sauer's lemma and the VC dimension of hyperplanes (see, e.g., theorem 5.5 in Schölkopf & Smola, 2002), the size is at most $O((|P_S|^2)^{d^*_{\Phi,S}})$. □

Theorem 3.

(i) For any Φ, $|\Theta| = O(|P_S|^{8(R/\mu^*)^2})$.

(ii) If $\mathcal{X} \subseteq \mathbb{R}^\ell$ and Φ is the identity mapping over P_S, then $|\Theta| = O(|P_S|^{\min\{8(R/\mu^*)^2, 2\ell\}})$.

(iii) If $\mathcal{X} \subseteq \mathbb{R}^\ell$ and Φ satisfies that $\big\langle \Phi(z), \Phi(x) \big\rangle$ is monotone decreasing with respect to $\|z - x\|_2$ (e.g., the mapping defined by the gaussian kernel) and $U = \{\Phi(z) \mid z \in \mathcal{X} \subseteq \mathbb{R}^\ell, \|\Phi(z)\|_{\mathbb{H}} \leq 1\}$, then $|\Theta| = O(|P_S|^{\min\{8(R/\mu^*)^2, 2\ell\}})$.

Proof. (i) We follow the argument in lemma 4. For the set of classifiers $F = \{f : \Phi_{\text{diff}}(P_S) \to \{-1, 1\} \mid f = \text{sign}(\langle \mathbf{u}, \mathbf{v} \rangle),\ \|\mathbf{u}\|_{\mathbb{H}} \leq 1,\ \min_{\mathbf{v} \in \Phi_{\text{diff}}(P_S)} |\langle \mathbf{u}, \mathbf{v} \rangle| = \mu\}$, its VC dimension is known to be at most R^2/μ^2 for $\Phi_{\text{diff}}(P_S) \subseteq \{\mathbf{v} \mid \|\mathbf{v}\|_{\mathbb{H}} \leq 2R\}$ (see, e.g., Schölkopf & Smola, 2002). By the definition of μ^*, for each intersection given by hyperplanes, there always exists a point $\mathbf{u}$ whose inner product between each hyperplane is at least μ^*. Therefore, the size of the intersections is bounded by the number of possible labelings in the dual space by $U'' = \{\mathbf{u} \in \mathbb{H} \mid \|\mathbf{u}\|_{\mathbb{H}} \leq 1, \min_{\mathbf{v} \in \Phi_{\text{diff}}(P_S)} |\langle \mathbf{u}, \mathbf{v} \rangle| = \mu^*\}$. Thus, we obtain that $d^*_{\Phi,S}$ is at most $8(R/\mu^*)^2$, and by lemma 4, we complete the proof of case i.

(ii) In this case, the Hilbert space $\mathbb{H}$ is contained in $\mathbb{R}^\ell$. Then, by the fact that VC dimension $d^*_{\Phi,S}$ is at most ℓ and lemma 4, the statement holds.

(iii) If $\left\langle \Phi(z), \Phi(x) \right\rangle$ is monotone decreasing for $\|z - x\|$, then the following holds:

$$\arg\max_{x \in \mathcal{X}} \left\langle \Phi(z), \Phi(x) \right\rangle = \arg\min_{x \in \mathcal{X}} \|z - x\|_2.$$

Therefore, $\max_{\mathbf{u}:\|\mathbf{u}\|_{\mathbb{H}}=1} \langle \mathbf{u}, \Phi(x) \rangle = \|\Phi(x)\|_{\mathbb{H}}$, where $\mathbf{u} = \frac{\Phi(x)}{\|\Phi(x)\|_{\mathbb{H}}}$. It indicates that the number of Voronoi cells made by $V(x) = \{z \in \mathbb{R}^{\ell} \mid z = \arg\max_{x \in B}(z \cdot x)\}$ corresponds to the $\hat{V}(x) = \{\Phi(z) \in \mathbb{H} \mid z = \arg\max_{x \in B} \langle \Phi(z), \Phi(x) \rangle\}$. Then, by following the same argument for the linear kernel case, we get the same statement. □

Now we are ready to prove theorem 2.

Proof of Theorem 2. By using lemmas 1, and 2, we obtain the generalization bound in terms of the gaussian complexity of H. Then, by applying lemma 3 and theorem 3, we complete the proof. □

B.1 Hyperparameter Tuning for Time-Series Classification. In the experiment for time-series classification, we roughly tuned ν and σ^2 of the gaussian kernel. As we mentioned before, we need high computation time when learning very large time series. The main computational cost is to iteratively solve weak learning problems by using an LP (or QP) solver. The number of constraints of the optimization problem 5.1 depends on the total number of instances in negative bags. Therefore, in the hyperparameter tuning phase, we finish solving each weak learning problem by obtaining the solution of the optimization problem 6.1. Using the rough weak learning problem, we tuned ν and σ through a grid search via three runs of 3-fold cross-validation.

Acknowledgments

This work is supported by JST CREST (grant JPMJCR15K5) and JSPS KAKENHI (grant JP18K18001). In the experiments, we used the computer resource offered under the category of General Projects by the Research Institute for Information Technology, Kyushu University.

References

Andrews, S., & Hofmann, T. (2004). Multiple instance learning via disjunctive programming boosting. In S. Thrun, L. K. Saul, & B. Schölkopf (Eds.), *Advances in neural information processing systems, 16* (pp. 65–72). Cambridge, MA: MIT Press.

Andrews, S., Tsochantaridis, I., & Hofmann, T. (2003). Support vector machines for multiple-instance learning. In S. Becker, S. Thrun, & K. Obermayer (Eds.), *Advances in neural information processing systems, 15* (pp. 577–584). Cambridge, MA: MIT Press.

Auer, P., & Ortner, R. (2004). A boosting approach to multiple instance learning. *Lecture Notes in Computer Science: Vol. 3201. Proceedings of the European Conference on Machine Learning* (pp. 63–74). Berlin: Springer.

Bagnall, A., Lines, J., Bostrom, A., Large, J., & Keogh, E. (2017). The great time series classification bake off: A review and experimental evaluation of recent algorithmic advances. *Data Mining and Knowledge Discovery*, *31*(3), 606–660.

Bartlett, P. L., & Mendelson, S. (2003). Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, *3*, 463–482.

Carbonneau, M.-A., Cheplygina, V., Granger, E., & Gagnon, G. (2018). Multiple instance learning: A survey of problem characteristics and applications. *Pattern Recognition*, *77*, 329–353.

Chen, Y., Bi, J., & Wang, J. Z. (2006). MILES: Multiple-instance learning via embedded instance selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *28*(12), 1931–1947.

Chen, Y., Keogh, E., Hu, B., Begum, N., Bagnall, A., Mueen, A., & Batista, G. (2015). *The UCR time series classification archive.* www.cs.ucr.edu/∼eamonn/timeseries_data/.

Cuturi, M. (2011). Fast global alignment kernels. In *Proceedings of the International Conference on Machine Learning* (pp. 929–936). New York: ACM.

Demiriz, A., Bennett, K. P., & Shawe-Taylor, J. (2002). Linear programming boosting via column generation. *Machine Learning*, *46*(1–3), 225–254.

Dietterich, T. G., Lathrop, R. H., & Lozano-Pérez, T. (1997). Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, *89*(1–2), 31–71.

Doran, G. (2015). *Multiple instance learning from distributions*. PhD diss., Case Western Reserve University.

Doran, G., & Ray, S. (2014). A theoretical and empirical analysis of support vector machine methods for multiple-instance classification. *Machine Learning*, *97*(1–2), 79–102.

Gärtner, T., Flach, P. A., Kowalczyk, A., & Smola, A. J. (2002). Multi-instance kernels. In *Proceedings of the International Conference on Machine Learning* (pp. 179–186). New York: ACM.

Grabocka, J., Schilling, N., Wistuba, M., & Schmidt-Thieme, L. (2014). Learning time-series shapelets. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 392–401). New York: ACM.

Grabocka, J., Wistuba, M., & Schmidt-Thieme, L. (2015). Scalable discovery of time-series shapelets. *CoRR*, abs/1503.03238.

Hills, J., Lines, J., Baranauskas, E., Mapp, J., & Bagnall, A. (2014). Classification of time series by shapelet transformation. *Data Mining and Knowledge Discovery*, *28*(4), 851–881.

Hou, L., Kwok, J. T., & Zurada, J. M. (2016). Efficient learning of timeseries shapelets. In *Proceedings of the AAAI Conference on Artificial Intelligence* (pp. 1209–1215). Palo Alto, CA: AAAI Press.

Karlsson, I., Papapetrou, P., & Boström, H. (2016). Generalized random shapelet forests. *Data Mining and Knowledge Discovery*, *30*(5), 1053–1085.

Keogh, E. J., & Rakthanmanon, T. (2013). Fast shapelets: A scalable algorithm for discovering time series shapelets. In *Proceedings of the International Conference on Data*

Mining (pp. 668–676). Philadelphia: Society for Industrial and Applied Mathematics.

Le Thi, H. A., & Pham Dinh, T. (2018). DC programming and DCA: Thirty years of developments. *Mathematical Programming*, *169*(1), 5–68.

Löning, M., Bagnall, A., Ganesh, S., Kazakov, V., Lines, J., & Király, F. J. (2019). *sktime: A unified interface for machine learning with time series*. arXiv:1909.07872.

Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2012). *Foundations of machine learning*. Cambridge, MA: MIT Press.

Platt, J. C., Cristianini, N., & Shawe-Taylor, J. (2000). Large margin DAGs for multiclass classification. In S. A. Solla, T. K. Leen, & K. Müller (Eds.), *Advances in neural information processing systems*, *12* (pp. 547–553). Cambridge, MA: MIT Press.

Renard, X., Rifqi, M., Erray, W., & Detyniecki, M. (2015). Random-shapelet: An algorithm for fast shapelet discovery. In *Proceedings of the IEEE International Conference on Data Science and Advanced Analytics* (pp. 1–10). Piscataway, NJ: IEEE.

Sabato, S., & Tishby, N. (2012). Multi-instance learning with any hypothesis class. *Journal of Machine Learning Research*, *13*(1), 2999–3039.

Sangnier, M., Gauthier, J., & Rakotomamonjy, A. (2016). Early and reliable event detection using proximity space representation. In *Proceedings of the International Conference on Machine Learning* (pp. 2310–2319).

Schökopf, B., & Smola, A. (2002). *Learning with kernels: Support vector machines, regularization, optimization, and beyond*. Cambridge, MA: MIT Press.

Shapiro, A. (2009). Semi-infinite programming, duality, discretization and optimality conditions. *Optimization*, *58*(2), 133–161.

Shimodaira, H., Noma, K.-i., Nakai, M., & Sagayama, S. (2001). Dynamic time-alignment kernel in support vector machine. In *Proceedings of the International Conference on Neural Information Processing Systems* (pp. 921–928). Cambridge, MA: MIT Press.

Tao, P. D., & Souad, E. B. (1988). *Duality in D.C. (difference of convex functions) optimization. Subgradient methods*. In K.-H. Hoffmann, J. Zowe, J.-B. Hiriat-Urruty, & C. Lemarechal (Eds.), *Trends in mathematical optimization* (pp. 277–293). Berlin: Springer.

Tavenard, R., Faouzi, J., & Vandewiele, G. (2017). tslearn: A machine learning toolkit dedicated to time-series data. https://github.com/rtavenar/tslearn.

Warmuth, M., Glocer, K., & Rätsch, G. (2008). Boosting algorithms for maximizing the soft margin. In J. C. Platt, D. Köller, Y. Singer, & S. T. Rowweis (Eds.), *Advances in neural information processing systems*, *20* (pp. 1585–1592). Cambridge, MA: MIT Press.

Wistuba, M., Grabocka, J., & Schmidt-Thieme, L. (2015). Ultra-fast shapelets for time series classification. *CoRR*, abs/1503.05018.

Ye, L., & Keogh, E. (2009). Time series shapelets: A new primitive for data mining. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 947–956). New York: ACM.

Yu, C.-N. J., & Joachims, T. (2009). Learning structural SVMs with latent variables. In *Proceedings of the International Conference on Machine Learning* (pp. 1169–1176). Omnipress.

Yu, F. X. X., Suresh, A. T., Choromanski, K. M., Holtmann-Rice, D. N., & Kumar, S. (2016). Orthogonal random features. In D. D. Lee, M. Sugiyama, U. V. Luxburg,

I. Guyon, & R. Garnett (Eds.), *Advances in neural information processing systems*, *29* (pp. 1975–1983). Red Hook, NY: Curran.

Zhang, C., Platt, J. C., & Viola, P. A. (2006). Multiple instance boosting for object detection. In Y. Weiss, B. Scholköpf, & J. C. Platt (Eds.), *Advances in neural information processing systems*, *18* (pp. 1417–1424). Cambridge, MA: MIT Press.

Zhang, D., He, J., Si, L., & Lawrence, R. (2013). MILEAGE: Multiple instance learning with global embedding. In *Proceedings of the International Conference on Machine Learning* (pp. 82–90). Omnipress.

Received November 13, 2019; accepted April 8, 2020.