Isolation Set-Kernel and Its Application to Multi-Instance Learning

Bi-Cun Xu
National Key Laboratory for Novel Software Technology,
Nanjing University
China
xubc@lamda.nju.edu.cn

Kai Ming Ting
School of Science, Engineering and Information Technology,
Federation University
Australia
kaiming.ting@federation.edu.au

Zhi-Hua Zhou
National Key Laboratory for Novel Software Technology,
Nanjing University
China
zhouzh@lamda.nju.edu.cn

ABSTRACT

Set-level problems are as important as instance-level problems. The core in solving set-level problems is: how to measure the similarity between two sets. This paper investigates data-dependent kernels that are derived directly from data. We introduce Isolation Set-Kernel which is solely dependent on data distribution, requiring neither class information nor explicit learning. In contrast, most current set-similarities are not dependent on the underlying data distribution. We theoretically analyze the characteristic of Isolation Set-Kernel. As the set-kernel has a finite feature map, we show that it can be used to speed up the set-kernel computation significantly. We apply Isolation Set-Kernel to Multi-Instance Learning (MIL) using SVM classifier, and demonstrate that it outperforms other set-kernels or other solutions to the MIL problem.

CCS CONCEPTS

• **Computing methodologies** → **Kernel methods**; *Support vector machines*;

KEYWORDS

Data-dependent kernel, Feature map, SVM classifiers, Multi-Instance Learning

ACM Reference format:
Bi-Cun Xu, Kai Ming Ting, and Zhi-Hua Zhou. 2019. Isolation Set-Kernel and Its Application to Multi-Instance Learning. In *Proceedings of The 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Anchorage, AK, USA, August 4–8, 2019 (KDD '19),* 9 pages.
https://doi.org/10.1145/3292500.3330830

1 INTRODUCTION

In many domains such as prediction and regression, set-level problems are as important as instance-level problems. One of the most important set-level problems is Multi-Instance Learning (MIL) [9]. Different from traditional single-instance learning, the input to an MIL learner is a dataset which consists of a number of bags, where bags have different numbers of instances. In addition, a bag is positive only if it contains at least one positive instance with other negative instances; otherwise it is a negative bag. Although the labels of the training bags are known, the labels of the instances in the bags are unknown.

Over the years, many MIL methods [2, 10, 11, 21, 22, 24–26] have been developed. All of these methods convert MIL problem into single-instance learning (SIL) problem either explicitly or implicitly. The simplest approach treats all instances in a bag to have the same label as the bag label; and then apply SIL methods to solve the problem [2, 10, 22, 24].

The kernel-based approach focuses on either designing a set-kernel based on the conventional kernel such as Gaussian kernel [11, 25], or using an existing set-kernel such as Fisher kernel to tailor to MIL [21]. Kernel-based methods usually have lower time complexity than non-kernel-based methods. This is because they can convert each bag into a mapped instance; while non-kernel-based methods need to deal with instances in the input space whose number is much larger than the number of bags.

We follow the kernel-based approach to tackle MIL. The proposed set-kernel differs from Gaussian kernel-based methods [11, 25] in two ways: (a) It is based on a *data-dependent* kernel called Isolation Kernel [19]; whereas Gaussian (or other related) kernels are *data independent*. (b) The feature map of the proposed set-kernel is a fixed length, sparse representation; whereas Gaussian or others have an infinite number of features.

The proposed set-kernel differs from the Fisher kernel-based method [21] in two ways: (i) it does not require a likelihood estimation method (used in Fisher kernel) or any other explicit learning. (ii) Fisher Kernel's feature map is a dense representation whereas the proposed set-kernel is a sparse representation.

A key challenge in set-level problems such as MIL is to design a good set-kernel to measure the similarity between any two sets.

The contributions of this paper are:

(1) Introducing a new data-dependent set-to-set kernel, called Isolation Set-Kernel, which adapts to the density structure of a given dataset.
(2) Theoretically analyzing the characteristic of the proposed set-kernel.
(3) Introducing a feature map of Isolation Set-Kernel, which converts a set of any size to a mapped instance of fixed length. This characteristic reduces the runtime of an algorithm significantly by reducing the input data set size.
(4) Applying a weighted version of the set-kernel to tackle Multi-Instance Learning (MIL).

KDD '19, August 4–8, 2019, Anchorage, AK, USA

ACM ISBN 978-1-4503-6201-6/19/08…$15.00
https://doi.org/10.1145/3292500.3330830

2 RELATED WORK

Two lines of research are directly related to our work. The first is data-dependent instance-level kernels and the second is set-kernels in MIL.

Isolation Kernel [19] is first proposed as a similarity between two instances which approximates well to a data independent kernel function called Laplacian kernel under uniform density distribution. With this revelation, Isolation Kernel can be viewed as a data-dependent kernel that adapts a data independent kernel to the density structure of a dataset.

Our work is an extension of Isolation Kernel from instance-level similarity measure to set-level similarity measure. We re-state its definition [19] as follows:

Let D be the given dataset; and $\mathcal{H}(D)$ denote the set of all partitioning H that are admissible under D where each isolation partition $\theta \in H$ isolates one instance x from the rest of the instances in a random subset $\mathcal{D} \subset D$, and $|\mathcal{D}| = \psi$.

Definition 2.1. For two instances $x, y \in \mathbb{R}^d$, Isolation Kernel of x and y wrt D is defined to be the expectation taken over the probability distribution on all partitioning $H \in \mathcal{H}_\psi(D)$ that both x and y fall into the same isolation partition $\theta \in H$:

$$K(x, y \mid D) = \mathbb{E}_{\mathcal{H}_\psi(D)}[\mathbb{I}(x, y \in \theta | \theta \in H)] \tag{1}$$

where $\mathbb{I}(B)$ is the indicator function which output 1 if B is true; otherwise $\mathbb{I}(B) = 0$.

In practice, Isolation Kernel [19] would be estimated from a finite number of partitionings $H_i \in \mathcal{H}_\psi(D), i \in 1, \ldots, t$ as follows:

$$K(x, y \mid D) = \frac{1}{t} \sum_{i=1}^{t} \sum_{\theta \in H_i} \mathbb{I}(x \in \theta)\, \mathbb{I}(y \in \theta) \tag{2}$$

Thus $K(x, y) \in [0, 1]$.

In the context of instance-level SVM classifiers, Isolation Kernel [19] has been shown to be more effective than existing approaches such as distance metric learning [20, 23], multiple kernel learning [12, 16] and Random Forest kernel [4, 8].

Both miGraph and MiGraph [25] map a bag to an undirected graph and use a graph-kernel to solve it. The graph kernel, which is also based on the Gaussian kernel, calculates the similarity of instances only if the individual instances are part of a graph. As a result, the graph kernel can be viewed as a set-kernel.

However, none of the above kernels, being based on Gaussian kernel, take the underlying data distribution into consideration; although some such as miGraph consider the structure in a bag.

Fisher Kernel [13] is a set-kernel used in miFV [21] for MIL. miFV learns a Gaussian mixture model (GMM) from SIL instances because the likelihood estimation is required for Fisher Kernel. After learning a GMM, each bag is converted to a mapped instance represented as a Fisher Vector. A linear classifier is trained from the mapped instances as the final classifier.

Two SVM-based methods, which do not design or utilize a set-kernel, tailor the standard objective function and constraints to MIL. mi-SVM [2] aims at finding a hyperplane between instances whose labels subjected to constraints defined by bags labels. Mi-SVM [2] aims at finding a hyperplane between bags whose labels contributed by instances.

A representative non-kernel-based method CCE [26] uses constructive clustering to re-represent a bag by d binary features, where the value of the i-th feature is set to one if the bag has at least one instance falling into the i-th cluster; and zero otherwise. Through repeating the above process with different values of d, many classifiers can be generated and then they can be combined into an ensemble for prediction.

3 ISOLATION SET-KERNEL

Let $S_i = \{x_1, ..., x_{w_i}\}$ be a dataset sampled from an unknown probability density function $S_i \sim F$, with $x_k \in \mathbb{R}^d$. Let $D = \{x_k \mid x_k \in S_i,\ S_i \in D, i = 1, \ldots, n\}$ be the overall dataset that is merged from all sets S_i. Let $\mathcal{H}(D)$ denote the set of all partitioning H that are admissible under D where each isolation partition $\theta \in H$ isolates one instance x_k from the rest of the instances in a random subset $\mathcal{D} \subset D$, and $|\mathcal{D}| = \psi$.

Definition 3.1. **ISK (Isolation Set-Kernel).** For any two sets S and T, ISK of S and T wrt D is defined to be the expectation over the probability distribution on all partitioning $H \in \mathcal{H}_\psi(D)$ that both $x \in S$ and $y \in T$ fall into the same isolation partition $\theta \in H$:

$$\mathcal{K}_\psi(S, T \mid D) = \mathbb{E}_{\mathcal{H}_\psi(D)}[\mathbb{I}(x, y \in \theta \mid \theta \in H; x \in T, y \in S)] \tag{3}$$

In practice, $\mathcal{K}_\psi$ is estimated from a finite number of partitionings $H_i \in \mathcal{H}_\psi(D)$, $i = 1, ..., t$ as follows.

$$\begin{aligned} \mathcal{K}_\psi(x, S \mid D) &= \frac{1}{t|S|} \sum_{y \in S} \sum_{i=1}^{t} \mathbb{I}(x, y \in \theta \mid \theta \in H_i) \\ &= \frac{1}{|S|} \sum_{y \in S} K(x, y \mid D) \end{aligned} \tag{4}$$

$$\mathcal{K}_\psi(S, T \mid D) = \frac{1}{|T|} \sum_{x \in T} \mathcal{K}_\psi(x, S \mid D) \tag{5}$$

The ISK has a feature map which has the following characteristic: a sparse representation with a user-definable number of features; in addition to be data-dependent. This is different from the feature maps of commonly used kernels such as RBF and polynomial kernels which are infinite or large number of features. The feature map is given as follows.

Let $\mathbf{v}(S|H)$ be a ψ-length vector representing all partitions $\theta_j \in H$, $j = 1, \ldots, \psi$ into which the proportion of instances in S fall. The j-component of the vector is defined as follows:

$$v_j(S|H) = \frac{1}{|S|} \sum_{y \in S} \mathbb{I}(y \in \theta_j \mid \theta_j \in H) \tag{6}$$

Definition 3.2. **Feature map of ISK.** For any set $S \subset \mathbb{R}^d$, the feature mapping $\Phi : S \to \mathbb{R}^{t \times \psi}$ of $\mathcal{K}_\psi$ is a vector that represents all the partitioning $H_i \in \mathcal{H}_\psi(D), i = 1, \ldots, t$ into which the proportion of $x \in S$ fall:

$$\Phi(S) = [\mathbf{v}(S|H_1), \mathbf{v}(S|H_2), \ldots, \mathbf{v}(S|H_t)] \tag{7}$$

$\Phi(S)$ is a fixed length vector, obtained through a concatenation operation of $\Phi_i(S)$ for $i = 1, \ldots, t$. Applying this feature map, set S of any size in $\mathbb{R}^d$ is converted to a mapped instance in $\mathbb{R}^{t\psi}$. Note that this feature map is a data-dependent, sparse representation.

Lemma 3.3. *$\mathcal{K}_\psi$ is a valid kernel, i.e., $\mathcal{K}_\psi(S,T) \equiv \langle \Phi(S), \Phi(T) \rangle$.*

Proof. Equation (5) can be re-expressed as follows:

$$\begin{aligned}
\mathcal{K}_\psi(S,T) &= \frac{1}{|T|} \sum_{x \in T} \left[\frac{1}{t|S|} \sum_{y \in S} \sum_{i=1}^{t} \sum_{\theta \in H_i} \mathbb{I}(x \in \theta) \mathbb{I}(y \in \theta) \right] \\
&= \frac{1}{t|T||S|} \sum_{i=1}^{t} \left[\sum_{x \in T} \sum_{y \in S} \sum_{\theta \in H_i} \mathbb{I}(x \in \theta) \mathbb{I}(y \in \theta) \right] \\
&= \frac{1}{t|T||S|} \sum_{i=1}^{t} \Phi_i(S)^\top \Phi_i(T) \\
&= \frac{1}{t} \Phi(S)^\top \Phi(T) \\
&= const \times \langle \Phi(S), \Phi(T) \rangle
\end{aligned} \tag{8}$$

□

The feature map of ISK has the following characteristics and their associated advantages:

(1) Because it is a fixed length representation, individual sets of different sizes are converted to individual mapped instances having the same feature size. This is a desirable characteristic which is very useful for many set-level problems.
(2) It adapts to the density structure of a given dataset, unlike existing data independent kernels such as Gaussian kernel.
(3) It is a sparse representation that can facilitate an efficiency kernel computation than the given data representation.

4 ISOLATION SET-KERNEL IMPLEMENTED WITH VORONOI DIAGRAM

There are currently two successful partitioning mechanisms to obtain the required partitions to yield Isolation Kernel. They are Isolation Forest [14, 19] and Voronoi diagram [3, 15]. In this paper, we use the Voronoi diagram to implement Isolation Set-Kernel.

Given a Voronoi diagram H constructed from sample $\mathcal{D}$ of ψ instances (randomly selected), the Voronoi cell centered at $z \in \mathcal{D}$ is:

$$\theta[z] = \{x \in \mathbb{R} \mid z = \arg\min_{z \in \mathcal{D}} d(x, z)\}$$

where $d(x, y)$ is Euclidean distance.

Definition 4.1. For any two sets $S, T \subset \mathbb{R}^d$, the Voronoi diagram version of ISK of S and T wrt D is defined to be the expectation over the probability distribution on all Voronoi diagrams $H \in \mathcal{H}_\psi(D)$ that both $x \in S$ and $y \in T$ fall into the same Voronoi cell $\theta \in H$:

$$\begin{aligned}
\mathcal{K}_\psi(S,T \mid D) &= \mathbb{E}_{\mathcal{H}_\psi(D)} [\mathbb{I}(x, y \in \theta[z] \mid \theta[z] \in H; x \in S, y \in T)] \\
&= \mathbb{E}_{\mathcal{D} \subset D} [\mathbb{I}(x, y \in \theta[z] \mid z \in \mathcal{D}; x \in S, y \in T)] \\
&= P(x, y \in \theta[z] \mid z \in \mathcal{D} \subset D; x \in S, y \in T)
\end{aligned}$$

where P denote the probability.

To demonstrate the distribution of ISK with Voronoi diagram, we use a set $S_0 : \{(0, 0.1), (0, -0.1), (0.1, 0), (-0.1, 0)\}$ on a 2-dimensional dataset with uniform density distribution. We use S_0 as the reference set; measure the similarity between S_0 and another set $S_\mu : \{(0+a, 0.1+b), (0+a, -0.1+b), (0.1+a, 0+b), (-0.1+a, 0+b)\}$, where $a \in [-1, 1], b \in [-1, 1]$. The resultant set-kernel distribution of ISK is shown in Figure 1(a). This distribution is similar to that of MI-Kernel [11], shown in Figure 1(b). MI-Kernel is one of the best *data independent* set-kernels for multi-instance learning thus far.

To contrast the data-dependent ISK with the data independent MI-Kernel, we use a dataset which has large density differences between two regions, where $x = 0$ is the boundary. The density ratio of region $x > 0$ and region $x < 0$ is $1 : 100$.

A comparison of kernel distributions of ISK and MI-Kernel [11] is shown in Figures 1(b) and 1(c) using this dataset. As MI-Kernel is a data independent kernel, it has the same kernel distribution independent of the data distribution. In contrast, ISK adapts to the local data distribution—see the difference in kernel distributions between Figures 1(a) and 1(c) as the data distribution changes from uniform density distribution to two-density-regions. This data dependency allows classifiers such as SVM to improve their predictive accuracy.

5 THEORETICAL CHARACTERISTIC OF ISOLATION SET-KERNEL

The Voronoi diagram has the required property of the space partitioning mechanism to produce large partitions in a sparse region and small partitions in a dense region. This yields the characteristic of ISK: **two sets (set-pair A) in sparse region are more similar than two sets (set-pair B) in dense region, if the pair-wise inter-point distance between the two sets of set-pair A is equal to those of set-pair B**:

Let $\rho(x)$ denote the density at instance x, we have the following statement based on definition 4.1:

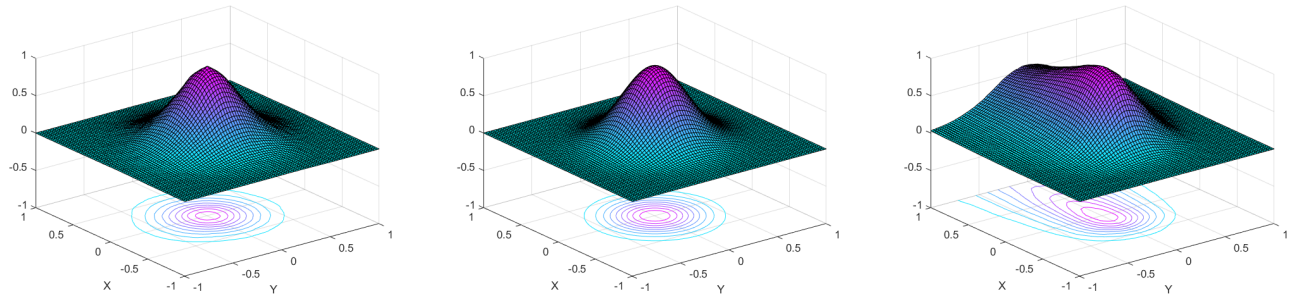
Theorem 5.1. *$\forall S, T \subset \mathcal{X}_\alpha$ (sparse region) and $\forall S', T' \subset \mathcal{X}_\beta$ (dense region) such that $\forall_{z \in \mathcal{X}_\alpha, z' \in \mathcal{X}_\beta}\ \rho(z) < \rho(z')$, the ISK $\mathcal{K}_\psi$ has the characteristic that,*

$$\begin{aligned}
&P(x, y \in \theta[z] \mid x \in S, y \in T) > P(x', y' \in \theta[z'] \mid x' \in S', y' \in T') \\
&\equiv \mathcal{K}_\psi(S, T \mid D) > \mathcal{K}_\psi(S', T' \mid D)
\end{aligned}$$

The proof follows the same argument used in a similar Lemma for Isolation Kernel [15]. The sketch of the proof is given as follows: (i) If two instances sampled from the two sets fall into the same Voronoi cell, then the distances of these individual instances to this cell centre must be shorter (or equal) than those to every other cell centre in a Voronoi diagram formed by all cell centres. (ii) In a subset of ψ instances, sampled from D, used to form a Voronoi diagram, the probability of two instances sampled from two sets falling into the same Voronoi cell can then be estimated based on the condition stated in (i). (iii) The probability of two instances sampled from two sets of equal pair-wise inter-point distance falling into the same Voronoi cell is a monotonically decreasing function wrt the density of the cells.

Proof. Let $V(x, y)$ be a local region covering both x and y as a ball centred at the middle between x and y, with $d(x, y)$ as the diameter of the ball. Assume that the density in $V(x, y)$ is uniform and denoted as $\rho(V(x, y))$.

Let $\mathcal{N}_\epsilon(x)$ be the ϵ-neighbourhood of x, i.e., $\mathcal{N}_\epsilon(x) = \{y \in D \mid d(x, y) \leq \epsilon\}$ The probability of both x and y are in the same Voronoi cell $\theta[z]$ is equivalent to the probability of an instance $z \in D$ being the nearest neighbour of both x and y wrt all other



(a) ISK under uniform density distribution (b) MI-Kernel ($\gamma = 8$) (c) ISK ($\psi = 32$)

Figure 1: ISK versus MI-Kernel for two sets S_0 and S_μ. The dataset used in (b) & (c) has the following data distribution: The density ratio at region $x > 0$ and at region $x < 0$ is 1:100.

instances in D, i.e., the probability of selecting $\psi - 1$ instances which are all located outside the region $U(x, y, z)$, where $U(x, y, z) = \mathcal{N}_{d(x,z)}(x) \cup \mathcal{N}_{d(y,z)}(y)$. To simplify notations, $z \in D$ is omitted. Then the probability of $x, y \in \theta[z] \mid z \in V(x, y), x \in S, y \in T) =$

$$\sum_{x_i \in S, y_j \in T} \left[P(x_i \mid x_i \in S) P(y_j \mid y_j \in T) P\left[z_1, z_2, ..., z_{\psi-1} \notin U(x_i, y_j, z)\right]\right]$$

For every $x_i \in S$ and $y_j \in T$:

$$P(x_i \mid x_i \in S) P(y_j \mid y_j \in T) P\left[z_1, z_2, ..., z_{\psi-1} \notin U(x_i, y_j, z)\right] \propto (1 - \frac{\mathbb{E}_{z \sim V(x_i, y_j)}\left[|U(x_i, y_j, z)|\right]}{|D|})^{(\psi-1)}$$

where $|W|$ denote the cardinality of W.

Assume $U(x_i, y_j, z)$ is also uniform distributed, with the same density $\rho(V(x_i, y_j))$, the expected value of $|U(x_i, y_j, z)|$ can be estimated as:

$$\mathbb{E}_{z \sim V(x_i, y_j)}\left[|U(x_i, y_j, z)|\right] = \mathbb{E}_{z \sim V(x_i, y_j)}\left[v(U(x_i, y_j, z))\right] \times \rho(V(x_i, y_j))$$

where $v(W)$ denotes the volume of W.

Thus, for every pair $x_i \in S$ and $y_j \in T$:

$$P(x_i, y_j \in \theta[z] \mid z \in V(x_i, y_j)) \propto (1 - \mathbb{E}_{z \sim V(x_i, y_j)}\left[v(U(x_i, y_j, z))\right] \times \frac{\rho(V(x_i, y_j))}{|D|})^{\psi-1}$$

Therefore, we obtain the following Lemma:

Lemma 5.2. *For every pair $x_i \in S$ and $y_j \in T$, $P(x, y \in \theta[z] \mid z \in V(x, y), x \in S, y \in T)$ is correlated positively to*

$$(1 - \mathbb{E}_{z \sim V(x_i, y_j)}\left[v(U(x_i, y_j, z))\right] \times \frac{\rho(V(x_i, y_j))}{|D|})^{\psi-1}$$

Given two pairs of sets from two different regions but of equal pair-wise inter-point distance as follows: $\forall S, T \in \mathcal{X}_\alpha$ (sparse region) and $\forall S', T' \in \mathcal{X}_\beta$ (dense region) such that $d(x, y)_{x \in S, y \in T} = d_{x' \in S', y' \in T'}(x', y')$.

We sample x, y from S, T and x', y' from S', T'. Assume that data are uniformly distributed in both regions, and we sample $z, z' \in \mathcal{D}$ from D such that $z \in V(x, y)$ and $z' \in V(x', y')$. We have

$$\mathbb{E}_{z \sim V(x,y), x \in S, y \in T}\left[v(U(x, y, z))\right] = \mathbb{E}_{z' \sim V(x', y'), x' \in S', y' \in T'}\left[v(U(x', y', z'))\right],$$

because the volume $v(V(x, y))$ is equal to that $v(V(x', y'))$, independent of the density of the region.

Suppose that we choose a sufficient large sample size ψ of $\mathcal{D}$ which contain instances from all $\{V(x_i, y_j) \mid x_i \in S, y_j \in T\}$ and $\{V(x'_i, y'_j) \mid x'_i \in S', y'_j \in T'\}$. When the data are uniformly distributed in $U(x, y, z) \in \mathcal{X}_\alpha$ and $U(x', y', z') \in \mathcal{X}_\beta$. Based on Lemma 5.2, we have:

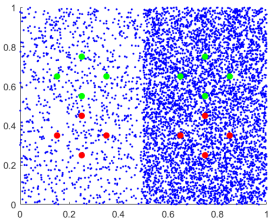
$$\begin{aligned} & P(x, y \in \theta[z] \mid z \in V(x, y), x \in S, y \in T) \\ > \; & P(x', y' \in \theta[z'] \mid z' \in V(x', y'), x' \in S', y' \in T') \\ \equiv \; & \mathcal{K}_\psi(S, T \mid D) > \mathcal{K}_\psi(S', T' \mid D) \end{aligned}$$

This means that $x' \in S'$ an $y' \in T'$ (in dense region) are more likely to be in different cells than $x \in S$ and $y \in T$ (in sparse region). □

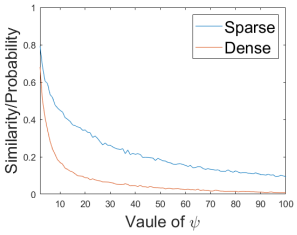
A simulation validating the above analysis is given in Figure 2. It compares $P(x, y \in \theta_\alpha \mid x \in S, y \in T)$ and $P(x', y' \in \theta_\beta \mid x' \in S', y' \in T')$ when S, T from a sparse region and S', T' from a dense region with equal pair-wise inter-point distance. Given fixed $\psi < |D|$ or fixed pair-wise inter-point distance, properties observed from Figure 2 are given as follows:

(1) $P(x, y \in \theta_\alpha \mid x \in S, y \in T) > P(x', y' \in \theta_\beta \mid x' \in S', y' \in T')$
(2) The rate of decrease $P(x, y \in \theta_\alpha \mid x \in S, y \in T)$ is faster that of $P(x', y' \in \theta_\beta \mid x' \in S', y' \in T')$, Thus $P(x, y \in \theta_\alpha \mid x \in S, y \in T)$ reaches 0 earlier.

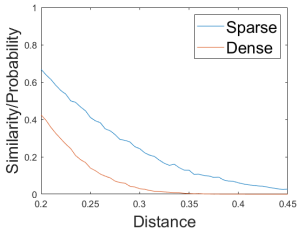
Isolation Set-Kernel is derived directly from data, requiring neither class information nor explicit learning. We show in the following section that how it can be applied to learn a classifier from a given training dataset with class labels in Multi-Instance Learning.



(a) Four reference sets of four points in two regions of different densities



(b) ψ increases.



(c) Inter-point distance increases

Figure 2: (a) Reference sets used in the simulation. The average position of the four instances in a set is the center of the set. (b) Results as ψ increases with inter-point distance=0.3. (c) Inter-point distance increases with $\psi = 12$. Inter-point distance between two sets is measured based on centers of the two sets.

6 APPLICATION TO MULTI-INSTANCE LEARNING

Multi-Instance Learning (MIL) is a set-level problem with a given dataset of bags with bag labels and unknown instance labels. The task is to learn a classifier from the given training set; then using the trained classifier to predict a label for each previously unseen bag.

Although instance labels are unknown, bag labels in MIL are defined as follows: A bag is a positive bag if it contains at least one positive instance; a bag is a negative bag only if all instances in the bag are negative.

Thus, positive instances are more important than negative instances, though the labels of instances are not available. In addition, the importance of an instance in a bag is different from another, even though they may have the same (unknown) labels. For example, an image is labeled as a garden. It may contains flowers, butterflies and the sky. The instances of the image are small patches on the image which constitute a bag of a certain concept. When only a small number of instances (i.e., a small bag) contributes to one concept such as flowers, these instances are more important than instances in a large bag of another concept (e.g., sky) because the former cannot be easily replaced by the latter. On the other hand, in a large bag of many instances, removing a few instances does not affect the label of the whole image.

We use this idea to incorporate instance weightings in ISK. This is described in the next three subsections.

6.1 Instance Weightings

Based on the intuition for MIL we described above, the number of instances in set S which are similar to x is used to weight x: the higher the number of similar instances in S, the lower the weight of x wrt S.

We measure similarity between two instances x and y based on Isolation Kernel $K(x, y)$ [19].

Definition 6.1. An instance y is ϵ-similar to x if $K(x, y) > \epsilon$, where $\epsilon \in [0, 1)$.

Definition 6.2. The instance weight of x wrt set S is defined to be the inverse of the number of instances in S which are ϵ-similar to x:

$$W_{Sx} = \frac{1}{\sum_{y \in T} \mathbb{I}(K(x, y) > \epsilon)} \tag{9}$$

As $x \in S$, the denominator has values in the range $[1, |S|]$.

To normalize the sum of weights of every bag to 1, we use the following equation:

$$\tilde{W}_{Sx} = \frac{W_{Sx}}{\sum_{y \in S} W_{Sy}} \tag{10}$$

6.2 Isolation Set-Kernel for Multi-Instance Learning

The weighted ISK is thus a weighted version of Equations 4 and 5:

$$\mathcal{K}_\psi(x, S \mid D) = \frac{1}{t|S|} \sum_{y \in S} \tilde{W}_{Sy} \sum_{i=1}^{t} \mathbb{I}(x, y \in \theta \mid \theta \in H_i) \tag{11}$$

$$\mathcal{K}_\psi(S, T \mid D) = \frac{1}{|T|} \sum_{x \in T} \tilde{W}_{Tx} \mathcal{K}_\psi(x, S \mid D) \tag{12}$$

where W_{Vz} is the weight of instance z in set V.

The j-component of the feature map of the weighted version of ISK is a modified version of Equation (6), i.e.,

$$v_j(S|H) = \frac{1}{|S|} \sum_{x \in S} \tilde{W}_{Sx} \times \mathbb{I}(x \in \theta_j \mid \theta_j \in H) \tag{13}$$

We normalize the kernel value to be in $[0, 1]$ as follows :

$$\tilde{\mathcal{K}}_\psi(S, T) = \frac{\mathcal{K}_\psi(S, T)}{\sqrt{\mathcal{K}_\psi(S, S)}\sqrt{\mathcal{K}_\psi(T, T)}} \tag{14}$$

6.3 Characteristic of The Weighted ISK

The weighted ISK has a characteristic stated as follows:

Proposition 6.3. *Two instances in dense region have more weights than two instances of equal inter-point distance in sparse region, provided that the number of other instances which are ϵ-similar to these two pairs of instances are the same.*

Proof. Given two instance-pairs $x, y \in S \subset \mathcal{X}_\alpha$ (sparse region), $x', y' \in T \subset \mathcal{X}_\beta$ (dense region). The weights of x and y are W_{Sx} and W_{Sy}; the weights of the x' and y' are $W_{Tx'}$ and $W_{Ty'}$.

When $\|x - y\| = \|x' - y'\|$, Isolation Kernel has the following characteristic [19]:

$$K(x, y) > K(x', y')$$

So there must exist a threshold ϵ that $K(x, y) > \epsilon > K(x', y')$.

Assume that the number of other instances in S which are ϵ-similar to pairs x, y is the same as that in T wrt x', y'. This gives

$$\frac{1}{\sum_{z \in S} \mathbb{I}(K(x,z) > \epsilon)} < \frac{1}{\sum_{z' \in S'} \mathbb{I}(K(x',z') > \epsilon)} \equiv W_{Sx} < W_{Tx'}$$

The same applies to $W_{Sy} < W_{Ty'}$. □

In other words, the weights of instances, as applied to ISK, are proportional to the density of the local region, under the stated assumption.

6.4 The Advantage of Feature Map with a Finite Number of Features

Recall from Definition 3.2 that the feature map of ISK is a fixed-length and sparse representation. Due to this characteristic, learning from the feature map of ISK reduces the time complexity of learning with the set-kernel significantly because the number of mapped instances is equal to the number of bags, rather than the total (original) instances of all bags. A mapped instance of bag S is a fixed-length vector $\Phi(S)$ as shown in Equation 7.

Existing set-kernels, such as MI-Kernel [11] that computes the sum of pair-wise Gaussian kernel between bags, are data independent and have feature maps with an infinite number of features.

For example, given two bags a and b with n_a and n_b instances respectively, MI-Kernel calculates the similarity between these two bags by computing a total of $n_a \times n_b$ Gaussian kernel calculations; while ISK calculates the similarity by 1×1 linear calculation only after converting a bag to a mapped instance.

Isolation set-kernel and the Fisher kernel used in miFV [21] have two similarities at high level and finer differences within each similarity: (i) both convert a bag to a fixed-length mapped instance. So, ISK and miFV has the same time complexity. However, Fisher kernel which relies on likelihood estimation using Gaussian Mixture Model does not appear to produce a good similarity measurement between sets. Our experiments in Section 7 show that Fisher kernel performs poorly in comparison with ISK. (ii) Both set-kernels are data-dependent. However, there is a key difference: The kernel characteristic of ISK is stipulated in Theorem 5.1; but the kernel characteristic of Fisher kernel is unclear.

6.5 Dealing with Text Datasets

A text dataset employs all words in the dictionary as attributes to represent a document, where each attribute indicates the number of times a word appears in the document. As a document contains only a small proportion of words in the dictionary, a large proportion of attributes have zero counts. An attribute with zero count means that the word it represents is irrelevant to a document.

This means attributes having zero values need to be treated differently from simple subtraction in distance calculation. For example, when two instances/documents which have zero value in attribute x_i, it doesn't means that they are exactly the same in attribute x_i.

To discount zero-value attributes in similarity measurement, we have adopted the method used previously in the context of bag-of-word representation in information retrieval [17]: ignore the attributes in which both instances have zero values.

The modified distance between x and y is computed as follows:

$$d(x,y) = \sqrt{\frac{\sum_{i \in C}(x_i - y_i)^2}{|C|}},\ C = \{i \mid x_i \neq 0 \vee y_i \neq 0\} \qquad (15)$$

The modified distance[1] is used to determine the nearest neighbour, where attributes are reduced locally based on relevant attributes which have values from the two instances under measurement. In other words, the Vonoroi diagram is based on locally relevant attributes only. As far as we know, this is the first time that the modified distance is used to partition the space based on Voronoi diagram.

7 EMPIRICAL EVALUATIONS OF ISOLATION SET-KERNEL IN MULTI-INSTANCE LEARNING

Here we evaluate the effectiveness of ISK's ability in adapting to the density structure of a given dataset in MIL.

The default settings of ISK are: the number of partitionings t is 200. The sampling size ψ is selected over $\psi \in \{2^m \mid m = 4, 5 \ldots 12\}$. ϵ is searched from $\{0.55, 0.6, \ldots 1.0\}$. For all methods using Gaussian kernel, the gamma is searched over $\{2^m \mid m = -5, \ldots 5\}$; and the threshold of miGraph is searched over $\{2^m \mid m = -5, \ldots 5\}$. All searches are conducted via a 5-fold cross-validation on the training set.

A total of nine MIL datasets are used. The statistics of all datasets are shown in Table 1. These include five benchmark datasets commonly used in multi-instance learning [2, 9]: Musk1, Musk2, Elephant, Fox and Tiger; two images datasets; one text dataset; and one large-scale dataset (speaker).

Table 1: Properties of datasets.

Dataset	#attribute	#bag			#instance
		#positive	#negative	total	
Elephant	230	100	100	200	1220
Fox	230	100	100	200	1320
Tiger	230	100	100	200	1391
Musk1	166	47	45	92	476
Musk2	166	39	63	102	6598
2000-Image	121	100	100	2000	7947
1000-Image	121	100	100	1000	4306
Text Dataset	200	50*20	50*20	100*20	80137
Speaker	20	190	240	430	583600

The datasets 2000-Image and 1000-Image [6, 7] contain twenty categories of COREL images, where each category has 100 images. Each image is regarded as a bag. The regions of interest in an image are regarded as instances. Images are segmented into regions such that each region is roughly homogeneous in color and texture.

We compared our methods with seven state-of-the-art methods: Three set-kernel based methods, i.e., MI-Kernel [11], miGraph [25], miFV [21]; and four non-set-kernel based methods: CCE [26], mi-SVM & Mi-SVM [2] and Simple-MI method [1]. Simple-MI method

[1] Any measure which makes use of distance can use Equation (15) as well. We apply this distance in the graph kernel of miGraph [25], but the accuracy decreases.

is a baseline which represents one bag with the mean vector of all the instances in the bag, and associates the mean vector with bag-level label to build a classifier. The top-level properties of these methods are summarized in Table 2.

Table 2: Properties of MIL methods. SVM* denotes that a modified optimization objective function is used. SVM denotes that a standard SVM with a designed kernel is used and trained from (original) instances. LinearSVM denotes that a standard SVM with linear kernel is used and trained from feature-mapped instances.

Algorithm	Category	Base Kernel	Classifier
ISK	Set-Kernel Based	Isolation Kernel	LinearSVM
MI-Kernel	Set-Kernel Based	Gaussian Kernel	SVM
miGraph	Set-Kernel Based	Gaussian Kernel	SVM
miFV	Set-Kernel Based	Fisher Kernel	LinearSVM
CCE	Ensemble	Linear Kernel	LinearSVM
mi-SVM	SVM Based	Gaussian Kernel	SVM*
Mi-SVM	SVM Based	Gaussian Kernel	SVM*
Simple-MI	Baseline	Gaussian Kernel	SVM

The performance of a method is measured in terms of predictive accuracy and CPU runtime. For each dataset, we report the average accuracy of 10-fold cross validation ad its standard errors.

All experiments are coded in Matlab and run on Intel i5 CPU and 16G RAM memory. For fairness of comparing runtime, all algorithms use the LIBSVM implementation [5].

7.1 Commonly Used Datasets

Table 3 shows the accuracy result on seven commonly used datasets in MIL. ISK achieves the best accuracy in most datasets. Note that there is no coincidence that the closest contenders to ISK are all set-kernel based methods: miGraph and miFV. ISK is better than these closest contenders on 6 out of 7 datasets; and it is better than other contenders on all 7 datasets. Our results show that miGraph is competitive with miFV, which is consistent with the results shown in [21].

The above result shows that ISK is the most effective data dependent set-kernel among existing set-kernels such as Fisher kernel (used in mvFV), miGraph and MI-Kernel. Its ability to adapt to local data distribution has directly contributed to its superior performance.

7.2 Text Categorization

Here we compare ISK with the three kernel-based methods (miGraph [25], MI-Kernel [11], miFV [21]) in text datasets which have a large proportion of zero-valued dimensions.

Twenty text categorization datasets [25] were derived from the 20 Newsgroups corpus popularly used in text categorization. Each news category has 50 positive and 50 negative bags. Each positive bag contains 3% instances randomly drawn from the target category; and negative instances are randomly and uniformly drawn from other categories. Average number of instances in each bag is around 40. Each instance is a post represented by the top 200 **TFIDF** features in all documents.

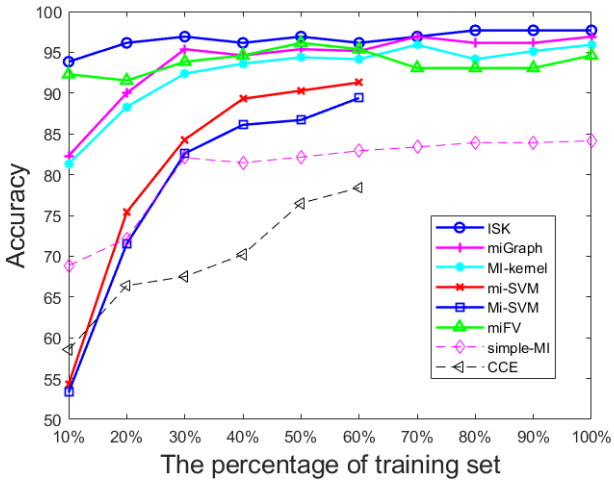


Figure 3: Classification results of the Speaker data set. CCE, miSVM and MI-SVM return no results in 48 hours when the percentage of training set is more than 60%.

The accuracy result is shown in Table 4. ISK achieves the best accuracy in most of these datasets. ISK is better than miGraph on 14 out of 20 datasets; and better than MI-Kernel and miFV on all 20 datasets.

It is interesting to note that the effectiveness of ISK is not affected by the zero ratio $\frac{n_v}{n_u}$, where n_v and n_u are the numbers of attributes having values and no values, respectively. In contrast, the accuracies of MI-Kernel and miFV were seriously impacted.

7.3 Large-Scale Dataset

To evaluate the performance of ISK in a large-scale dataset, we use the speaker dataset [1] which has a total of more than half a million instances. The training set has 120 positive and 120 negative bags; the validation set has 30 positive and 30 negative bags; and the testing set has 40 positive and 90 negative bags. The validation set is used for searching the parameters. We sample different ratios of the training set ranging from 10% to 100%.

Figure 3 shows that ISK achieves the best accuracy at all percentages of the training set; and it has high accuracy with even only 10% of the training set. The closest contenders are miGraph, MI-Kernel and miFV.

7.4 Runtime on Large-Scale Dataset

Here we compare the runtimes and time complexities of ISK and its Feature Map, together with those of its three closest contenders.

Finding the nearest neighbours with the Voronoi diagrams costs $O(\psi t n)$, where ψ, t, n are the numbers of instances in a subsample, partitionings in $\mathcal{H}$, and data size, repsectively. To compute the a $n \times n$ similarity matrix of ISK, it costs $O(n^2)$. Using ISK as a set-kernel, the training and testing of SVM cost $O(b^2)$. So the total time complexity of using ISK in SVM is $O(n^2 + b^2)$.

Table 3: Accuracy on 7 benchmark MIL datasets. Methods with the highest average accuracy of each column is marked in bold.

Algorithm	Elephant	Fox	Tiger	Musk1	Musk2	2000-Image	1000-Image
ISK	**89.2 ±0.6**	**61.5±0.7**	**82.6±0.7**	**89.9±0.9**	85.1±1.1	**75.2±0.6**	**85.1±1.1**
MI-Kernel	84.2±0.8	60.6±0.6	80.9±1.0	88.1±1.3	85.0±1.2	73.2±0.9	83.7±1.5
miGraph	85.3±0.6	61.3±0.8	81.6±0.8	88.7±1.1	**87.8±1.3**	72.1±0.7	84.5±1.0
miFV	84.1±0.8	60.7±0.9	81.3±0.6	87.8±1.1	86.8±1.3	71.4±0.6	83.7±1.2
CCE	79.5±1.2	59.1±1.3	76.9±1.1	84.1±1.4	71.2±1.7	62.8±1.3	69.3±1.5
mi-SVM	82.1±0.6	58.2±0.5	78.8±0.9	87.4±0.7	83.6±1.0	67.4±1.1	78.3±1.3
Mi-SVM	81.3±1.2	59.4±0.9	79.8±1.1	77.8±0.8	84.1±0.9	66.8±1.1	76.4±1.3
Simple-MI	81.8±0.8	57.7±0.5	79.2±0.6	85.7±0.4	83.2±1.2	63.7±0.6	70.5±0.8

Table 4: Accuracy on text categorization datasets.

Data set	ISK	miGraph	MI-Kernel	miFV	zero ratio
alt.atheism	**83.2±1.0**	82.1±1.1	56.5±1.7	71.4±0.9	0.0202
comp.graphics	**85.3±0.9**	84.2±0.9	51.7±1.3	57.2±0.6	0.0096
comp.os.ms	**67.5±1.6**	67.4±1.3	49.6±2.1	55.6±0.4	0.0149
comp.sys.ibm	76.7±1.5	**78.3±0.8**	53.1±1.6	57.7±1.8	0.0157
comp.sys.mac	80.8±2.1	**83.0±1.9**	50.8±2.2	54.1±0.6	0.0135
comp.windows.x	**82.4±1.0**	80.6±1.1	51.7±1.3	66.0±0.5	0.0126
misc.forsale	**70.1±1.3**	69.3±1.4	55.3±1.7	62.8±1.1	0.0230
rec.autos	79.9±1.7	**84.6±1.7**	52.7±1.9	59.7±1.1	0.0102
rec.motorcycles	**84.1±1.4**	82.7±1.3	59.3±1.3	74.8±1.4	0.0134
rec.sport.baseball	86.3±1.6	**88.8±1.5**	54.0±2.1	78.2±1.8	0.0118
rec.sport.hockey	85.5±1.7	**90.1±1.2**	51.2±1.8	79.5±1.0	0.0118
sci.crypt	**80.6±1.3**	76.8±1.5	54.6±1.4	61.3±2.1	0.0150
sci.electronics	**93.3±0.8**	92.9±0.7	54.7±1.7	52.8±0.9	0.0095
sci.med	**83.8±1.8**	83.7±1.6	50.3±1.8	75.3±1.9	0.0116
sci.space	**83.3±1.7**	83.1±1.5	54.4±1.9	74.5±1.8	0.0109
sci.religion	**83.7±1.1**	82.5±1.0	52.7±1.4	65.2±1.4	0.0220
talk.politics.guns	**79.7±1.7**	78.2±1.9	52.8±1.5	64.6±0.7	0.0157
talk.politics.mideast	**82.9±1.6**	81.6±1.3	53.6±1.7	73.7±1.0	0.0136
talk.politics.misc	70.8±1.5	**74.4±1.4**	51.7±1.8	70.6±1.6	0.0220
talk.religion.misc	**81.1±1.3**	76.2±1.4	53.8±1.4	73.4±1.3	0.0215
win/tie/loss	—	6/0/14	0/0/20	0/0/20	
Average Rank	1.30	1.70	3.95	3.05	

Table 5: Accuracy and Runtime (CPU seconds) on Speaker.

	FM of ISK	ISK	miGraph	MI-Kernel	miFV
preprocess	84	56	10	0.1	6
train	7	1103	1215	1288	4
test	6	1450	1117	1077	4
total	97	2609	2342	2365	14
accuracy	97.7	97.7	96.9	95.6	94.6

To transform to its feature map, the preprocessing costs $O(n)$. It costs $O(b^2)$ to calculate the dot product of two mapped vectors over all bags, where b is the number of bags in the given dataset. The total cost of learning with its feature map costs $O(n + b^2)$.

Preprocessing of the feature map involves creating the partitionings and the mapping from x to $\Phi(x)$; whereas preprocessing of ISK as a set-kernel involves creating the partitionings only.

The conversion from the given data representation to the feature map of ISK takes longer than the total time of training and testing using the feature map. This is because hundreds of partitions are used in the calculation. As each partition can be calculated independently, the computation can be easily parallelized to significantly reduce the runtime. This is amenable to GPU acceleration because it can be implemented in almost pure matrix manipulations, as shown in [15]. Training and testing using the feature map are fast. This is because, after converting each bag into a mapped instance, the total number of mapped instances is equal to the total number of bags, i.e., hundreds of mapped instances as opposed to half a million instances. This is made possible only because the feature map is in a specific form of sparse representation.

To show the advantage of the feature map of ISK, we use the largest dataset Speaker. The dataset is split into three parts, which is the same as in Section 7.3.

Table 5 shows that using the feature map of ISK ran one order of magnitude faster than that using ISK as a set-kernel. It has the same time complexity $O(n + b^2)$ as miFV; but the latter has the worst accuracy among the four algorithms.

7.5 Condition Under Which ISK Performs Well

To explore the condition under which ISK performs well in real datasets, we focus on the bags which are misclassified by SVM classifiers. For each misclassified bag, its nearest bag of a different bag label is identified, which is calculated by averaging pair-wise Euclidean distance of instances in the bags. Then, the density [2] of each instance is computed.

Let ρ_{iu} be the density of u-th instance in i-th bag, and $\rho_{\bar{i}v}$ be the density of v-th instance in i-th bag's nearest bag of a different bag label. For each pair of bags i and $\bar{i}$, we calculate maximum density—$max_{u,v}(\rho_{iu}, \rho_{\bar{i}v})$, minimum density—$min_{u,v}(\rho_{iu}, \rho_{\bar{i}v})$, density ratio—$max(\rho_{\bar{i}v})/min(\rho_{iu})$, of all pairs of instances between these two bags.

Table 6 shows the average of these values of all pairs of bags.

It is interesting to note that sets which are misclassified by MI-Kernel only have the highest density ratio; and those by ISK only have the lowest ratio on every dataset. This result indicates that, in

[2] To estimate the density of an instance on a dataset, we use kernel density estimation [18], where a multi-variate Gaussian is applied as the kernel function, and the bandwidth for dimension i is set to $b_i = \sigma_i \{\frac{4}{(d+2)n}\}^{1/(d+4)}$, where σ_i is the standard deviation of dimension i, d is the number of dimensions, and n is the dataset size.

Table 6: Densities and density ratios of SVM misclassified bags. The number in bracket is the total number of misclassified testing bags, obtained from ten-fold cross validation.

Dataset	Misclassified by	Maximum	Minimum	Ratio
Elephant	ISK (9)	0.7193	0.5040	1.3404
	both kernel (13)	0.7143	0.4872	1.6621
	MI-Kernel (21)	0.7183	0.4656	2.3831
Tiger	ISK (13)	0.6392	0.5214	1.6662
	both kernel (18)	0.6405	0.4617	2.0723
	MI-Kernel (15)	0.6431	0.3429	3.6084
Musk1	ISK (3)	0.4104	0.2817	1.4694
	both kernel (6)	0.4094	0.2795	1.8518
	MI-Kernel (5)	0.4111	0.2827	1.8976

SVM classifications, ISK produces better predictive accuracy than MI-Kernel in regions where class density varies hugely between classes. This result is consistent with the result of a similar experiment examining the condition under which Isolation Kernel is better than Laplacian kernel in SVM classifiers [19].

8 CONCLUSIONS

We introduce a new data-dependent set-kernel called Isolation Set-Kernel (ISK) and demonstrate its superiority in comparison with existing set-kernels which are data independent as well as data-dependent. The characteristics of ISK and weighted ISK—which are due to a specific form of data dependency—are stipulated; and their proofs are provided.

To apply to Multi-Instance Learning (MIL), we have introduced a weighted version of ISK that derives the weights of individual instances from their similarities wrt the set an individual instance belongs. The similarity depends on data distribution of the entire dataset as well as individual instances in a set. This form of data dependency is its key difference from existing set-kernels used in MIL. Our empirical evaluations show that ISK achieves high accuracy which is better than existing set-kernels or other MIL methods, all using SVM as the classifiers, on several multi-instance classification tasks.

ISK has a finite feature map with sparse representation—the key reason which enables conversion from MIL to an effective mapped SIL representation that significantly reduces the runtime because the number of mapped instances is equal to the number of bags, rather than the total number of (original) instances of all bags. Existing set-kernels either rely on conventional data independent kernels such as Gaussian kernel that have a feature map with an infinite number of features, or Fisher kernel which relies on a likelihood estimation. Both have prevented them from producing a feature mapping which is both efficient and effective.

9 ACKNOWLEDGEMENTS

Yu-Feng Li and Peng Zhao provided helpful comments in the initial draft. This research was supported by the National Key R&D Program of China (2018YFB1004300), NSFC (61751306), 111 Program (B14020), and Collaborative Innovation Center of Novel Software Technology and Industrialization. This material is based upon work partially supported by the Air Force Office of Scientific Research, Asian Office of Aerospace Research and Development (AOARD) under award number: FA2386-18-1-4032 (Kai Ming Ting).

REFERENCES

[1] Jaume Amores. 2013. Multiple instance classification: Review, taxonomy and comparative study. *Artificial Intelligence* 201, 4 (2013), 81–105.
[2] Stuart Andrews, Ioannis Tsochantaridis, and Thomas Hofmann. 2003. Support vector machines for multiple-instance learning. In *Advances in neural information processing systems*. 577–584.
[3] Franz Aurenhammer. 1991. Voronoi diagrams-a survey of a fundamental geometric data structure. *ACM Computing Surveys (CSUR)* 23, 3 (1991), 345–405.
[4] L. Breiman. 2000. Some infinity theory for predictor ensembles. *Technical Report 577* (2000).
[5] Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)* 2, 3 (2011), 27.
[6] Yixin Chen, Jinbo Bi, and James Ze Wang. 2006. MILES: Multiple-instance learning via embedded instance selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28, 12 (2006), 1931–1947.
[7] Y. Chen, J. Li, and J. Z. Wang. 2004. *Categorization by Learning and Reasoning with Regions*. Springer US. 99–121 pages.
[8] A. Davies and Z. Ghahramani. 2014. The random forest kernel and creating other kernels for big data from random partitions. *arXiv:1402.4293* (2014).
[9] Thomas G Dietterich, Richard H Lathrop, and Tomás Lozano-Pérez. 1997. Solving the multiple instance problem with axis-parallel rectangles. *Artificial intelligence* 89, 1-2 (1997), 31–71.
[10] Eibe Frank and Xin Xu. 2003. Applying propositional learning algorithms to multi-instance data. (2003).
[11] Thomas Gartner, Peter A. Flach, Adam Kowalczyk, and Alex J. Smola. 2002. Multi-Instance Kernels. In *Nineteenth International Conference on Machine Learning*. 179–186.
[12] Mehmet Gönen and Ethem Alpaydın. 2011. Multiple kernel learning algorithms. *Journal of machine learning research* 12, Jul (2011), 2211–2268.
[13] Tommi Jaakkola and David Haussler. 1999. Exploiting generative models in discriminative classifiers. In *Advances in neural information processing systems*. 487–493.
[14] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. 2008. Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*. IEEE, 413–422.
[15] Xiaoyu Qin, Kai Ming Ting, Ye Zhu, and Vincent Cheng Siong Lee. 2019. Nearest-Neighbour-Induced Isolation Similarity and Its Impact on Density-Based Clustering. In *Proceedings of The Thirty-Third AAAI Conference on Artificial Intelligence*.
[16] Alain Rakotomamonjy, Francis R Bach, Stéphane Canu, and Yves Grandvalet. 2008. SimpleMKL. *Journal of Machine Learning Research* 9, Nov (2008), 2491–2521.
[17] Gerard Salton and Michael J McGill. 1986. Introduction to modern information retrieval. (1986).
[18] Bernard W Silverman. 2018. *Density estimation for statistics and data analysis*. Routledge.
[19] Kai Ming Ting, Yue Zhu, and Zhi-Hua Zhou. 2018. Isolation kernel and its effect on SVM. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2329–2337.
[20] Fei Wang and Jimeng Sun. 2015. Survey on distance metric learning and dimensionality reduction in data mining. *Data Mining & Knowledge Discovery* 29, 2 (2015), 534–564.
[21] Xiu-Shen Wei, Jianxin Wu, and Zhi-Hua Zhou. 2017. Scalable algorithms for multi-instance learning. *IEEE transactions on neural networks and learning systems* 28, 4 (2017), 975–987.
[22] Xin Xu and Eibe Frank. 2004. Logistic regression and boosting for labeled bags of instances. In *Pacific-Asia conference on knowledge discovery and data mining*. Springer, 272–281.
[23] Pourya Zadeh, Reshad Hosseini, and Suvrit Sra. 2016. Geometric mean metric learning. In *International Conference on Machine Learning*. 2464–2471.
[24] Qi Zhang and Sally A Goldman. 2002. EM-DD: An improved multiple-instance learning technique. In *Advances in neural information processing systems*. 1073–1080.
[25] Zhi-Hua Zhou, Yu-Yin Sun, and Yu-Feng Li. 2009. Multi-instance learning by treating instances as non-iid samples. In *Proceedings of the 26th annual international conference on machine learning*. ACM, 1249–1256.
[26] Zhi Hua Zhou and Min Ling Zhang. 2007. Solving multi-instance problems with classifier ensemble based on constructive clustering. *Knowledge and Information Systems* 11, 2 (2007), 155–170.