Instance Selection for MI-Support Vector Machines

Bokyung Amy Kwon

Department of Artificial Intelligence, College of Computing, Hanyang Univeristy, ERICA, 15588, Korea
Email: amykwon@hanyang.ac.kr

Manuscript received November 15, 2024; revised December 10, 2024; accepted December 20, 2024; published January 21, 2025

Abstract—**Support Vector Machines (SVM) is a well-known algorithm in machine learning due to its superior performance, and it also functions well in Multiple-Instance (MI) problems. Our study proposes a schematic algorithm for selecting instances based on Hausdorff distance, which can be adapted to SVMs as input vectors under MI setting. We confirmed that SVMs in MI settings when using this instance selection strategy outperformed original approaches based on experiments with five benchmark datasets. In addition, Task Execution Times (TETs) were reduced by more than 80% based on MissSVM. Therefore, it is noteworthy to consider this representation adaptation for SVMs in MI setting.**

Keywords—**support vector machine, Margin, Hausdorff distance, representation selection, multiple-instance learning, machine learning**

I. Introduction

Support Vector Machines (SVMs) have consistently received significant attention in the field of machine learning. Their superior performance is particularly recognized due to their generalization ability, which results from structural risk minimization. The technique is therefore considered robust, especially when dealing with the increased dimensionality of input vectors [1].

MI-learning is a distinct area due to its hierarchical data structure and ambiguous label information. However, SVMs still work well primarily under the standard assumption that a positive bag contains at least one positive example while a negative bag consists of only negative examples. To satisfy this assumption, SVMs for MI-learning typically operate by assigning the same label information to the examples as to the entity or by treating missing values. Well-known algorithms in this area include mi-SVM, MI-SVM and MissSVM. These algorithms are notable because SVMs have been well adapted to the MI framework demonstrating comparable performance. However, the unbalanced and heterogeneous data characteristics remain a challenge.

Our study focuses on these complex data structures, and proposes an algorithmic scheme for instance selection to achieve fast adaptation to SVM-type algorithms in MI-learning without altering the original algorithm. To do this, we selected instances by slightly modifying Hausdorff distance to capture instances from each polarity collection, as its Max-Min operation is somewhat similar to the mechanism of SVMs. The rest of the paper is organized as follows. The next section briefly summarizes the related work and representation selection scheme in the Method section. In the Experiments and results, we evaluate our approach on five different benchmark datasets and compare task execution times (TETs) based on MissSVM. The results are described in detail. Finally, we discuss the strengths and weaknesses in the Discussion section.

II. Related Works

The learnability of Multiple Instance Learning (MI-learning) has primarily been studied in the classification of an entity unit based on the given label information [1, 2]. As it might be expected, no single approach has dominated in terms of performance under this framework, but various derivative forms of Support Vector Machines (SVM) have been developed to solve the problem under the ambiguity [3, 4]. Two transduction-based algorithms - mi-SVM and MI-SVM [3] are well-known approaches for applying SVM to the MI-learning framework. In these methods, SVMs are repeatedly applied to the examples based on their label information, treating them as belonging to the same bags while all label information remains constant. The only difference between mi-SVM and MI-SVM lies in the definition of population for learning: MI-SVM makes use of the most positive examples to retrieve information while mi-SVM utilizes all possible examples for learning. Additionally, MissSVM is another derivative form of SVM developed for MI-learning [5, 6]. This algorithm treats the label information of the examples in positive bags as missing values while those in negative bags are treated as negative. Then, it maximizes the margin on both labeled and unlabeled data by implementing SVM under the semi-supervised learning framework. As a result, it requires relatively more intensive computation time for optimization.

Additionally, embedding-based approaches were also developed. Diverse-Density (DD) [7] might be the first approach based on instance-based embedding. Ultimately, this algorithm seeks the concept by the most likely estimator expecting that a higher proportion of positive bags will be centered around it. Under the same framework, Expectation Maximization (EM)-DD [8] systematically updates the most-likely estimator using the Quasi-newton algorithm. Similarly, a graph-based approach was also developed by transforming an entity into a graph-like embedding, known as mi-graph [9]. In this approach, the entity is constructed by sets of instances, which releases the independence assumption. This approach also implements SVM on the similarity based on a graph-kernel. Finally, Multiple-Instance Learning via Embedded Instance Selection (MILES) [10] also applies 1-norm SVM to the feature space, which is formed by maximal similarity. This approach enabled us to classify examples more effectively. According to prior studies, it is clear that SVM plays an important role in classification under the MI-learning setting. Recently, a Distance-Aware Self-Attention-Based Model (DAS-MIL) was proposed to handle images under the same setting, which takes into account the spatial relationship between patches, demonstrating fair performance [11]. Additionally, although it is not a fully supervised learning framework, proximal SVM demonstrates efficiency under the MI setting [12].

doi: 10.18178/ijml.2025.15.1.1172

III. Methods

A. Standard Assumption

The problem can be defined as the standard assumption initially. The training set consists of pairs (B_i, Y_i) for i=1,…, n, where B_i and Y_i represent the i_{th} bag and its corresponding label, respectively. Each B_i consists of a set of instances, i.e., $B_i = \{B_{i1}, \dots, B_{in_i}\}$ where $B_{ij} \in R^p$, and if y_{ij} is the label of B_{ij}, Y_i is defined as follows.

$$y_i = \begin{cases} 1 & if\ \exists\, y_{ij}: y_{ij} = +1 \\ -1 & if\ \forall\, y_{ij}: y_{ij} = -1 \end{cases} \qquad (1)$$

(For simplicity, the collections of bags whose labels are +1, and -1 are denoted as B^+ and B^-, respectively.)

B. Brief Overview of SVMs

If the case is linearly separable, the SVM algorithm seeks the separating hyperplane with the largest margin. When all data points in the training set satisfy the following constraint [1]:

$$y_i \cdot (x_i \cdot \omega + b) - 1 \geq 0, \ \ for\ \forall i \qquad (2)$$

Then, the problem becomes minimizing the *Lagrangian* of Eq. (4), with respect to ω and b, subject to $\alpha_i \geq 0$,

$$L_p = \frac{1}{2}\|\omega\|^2 - \sum_{i=1}^{l} \alpha_i y_i (x_i \cdot \omega + b) + \sum_{i=1}^{l} \alpha_i \qquad (4)$$

Since L_p can be transformed into a dual form, it quickly becomes the maximization problem as shown in Eq. (5),

$$L_D = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j x_i \cdot x_j \qquad (5)$$

If $\alpha_i > 0$, the corresponding data points become support vectors, which lie on one of the hyperplanes.

C. Selection Scheme for Representation

One of the difficulties of MI-learning arises from the complex structure of the data as well as ambiguous label information. As a solution, our study proposes a representation selection method for each bag. In particular, when SVM or its derivatives are implemented as classifiers for MI-learning, this selection procedure proves to be especially useful and simplifies the structure.

Our study proposes a balanced instance selection method to represent an entity based on the Hausdorff distance. The Hausdorff distance uniquely measures the difference between two metric spaces for two non-empty subsets, A and B, where r is a positive real number, as shown in Eq. (6) [13].

$$d_H(A, B) = \inf\{r: A \subset U_r(B)\ and\ U_r(A) \supset B\} \qquad (6)$$

This original definition is well interpreted for different two vectors in various applications, and generalized to express the distance between any two sets using the 'Max-Min' operation as shown in Eq. (7), where $d(a-b)$ can represent any distance metric [14].

$$d_H(A, B) = \max\left(d_H(A, B)\ and\ d_H(B, A)\right) \qquad (7)$$

where

$$d_H(A, B) = \max_{a \in A}\left(\min_{b \in B} d(a-b)\right)$$

and

$$d_H(B, A) = \max_{b \in B}\left(\min_{a \in A} d(b-a)\right)$$

The 'Max-Min' operation represents the maximum distance any component of either set must travel to reach the other set as quickly as possible, resembling the definition of support vectors in SVM. In the MI-setting, if positivity is characterized by a single positive instance rather than multiple negative instances, the positive instance may be distant from the other negative instances. Therefore, it may not always be the best strategy to choose the closest instance to represent the entity. Based on this idea, we modified the Hausdorff distance metric for representation selection. (The pseudocode for the selection procedure is summarized in Algorithm 1.)

Suppose that the collections of positive and negative entities are denoted as B^+ and B^-, respectively. Then, Eq. (7) can be adapted to our study as follows:

$$d_H(B_i, B^+) = \max\left(d_H(B_i, B^+)\ and\ d_H(B^+, B_i)\right) \qquad (8)$$

Similarly,

$$d_H(B_i, B^+) = \max_{b_{ij} \in B_i}\left(\min_{b \in B^+} d(b_{ij} - b)\right)$$

but

$$d_H^*(B^+, B_i) = \min_{j | b \in B^+}\left(d_H(B^+, B_i)\right)$$

In this procedure, one instance is selected based on the Hausdorff distance metric as it is, while the other, $d_H^*(B^+, B_i)$, is selected as the matched counterpart from the other set. This matching is determined using the conventional Hausdorff distance metric. The final selection is made by taking the maximum of these two distances. Similarly, B^+ can be replaced with B^-. That is, each entity consists of a pair of instances representing opposite polarities, respectively. (The selection scheme is illustrated in Fig 1. The left plot shows the distribution of bags, while the middle and the right plot show the instances selected based on their distance from the sets with opposite polarities.)

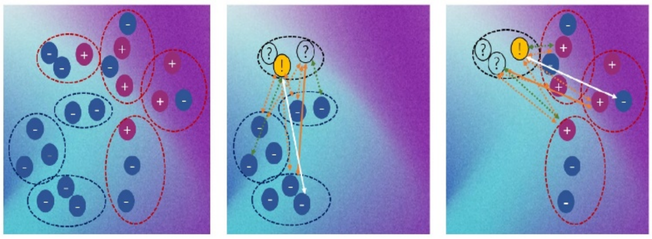

Fig. 1. An algorithmic selection using Hausdorff distance.

Algorithm 1. algorithmic procedure: Pseudo code for selection

```
Procedure:
Require: $D_t = \{(B_i, Y_i) | i = 1, \dots, n\}$ where $B_i \in \{B^+, B^-\}$

for $B_i \in D_t$ do:
    $\{B^+, B^-\} \leftarrow \{B^+ \setminus B_i,\ B^- \setminus B_i\}$
    \* compute H($B_i, B^+$) *\
        $d_H(B_i, B^+) = \max_{b_{ij} \in B_i}\left(\min_{b \in B^+} d(b_{ij} - b)\right)$
    where
        $d_H(B_i, B^+) = \max\left(d_H(B_i, B^+)\ and\ d_H(B^+, B_i)\right)$
        $d^*_H(B^+, B_i) = \min_{j|b \in B^+}(d_H(B^+, B_i))$

    \* compute H($B_i, B^-$) *\
        $d_H(B_i, B^-) = \max_{b_{ij} \in B_i}(\min_{b \in B^+} d(b_{ij} - b))$
    where
        $d_H(B_i, B^-) = \max(d_H(B_i, B^-)\ and\ d_H(B^-, B_i))$
        $d^*_H(B^-, B_i) = \min_{j|b \in B^+}(d_H(B^-, B_i))$

    select (j, j′) ← $\max_{j \in B_i} H(B_i, B^+), \max_{j\prime \in B_i} H(B_i, B^-)$
end for
```

D. *Fast Adaptation for SVMs*

The objective function of SVMs fundamentally seeks to maximize the margin based on support vectors. The selected instances are chosen from the collections of each polarity within the entity. Therefore, we hypothesize that this selection will enable SVMs to work more efficiently and reduce the computational burden.

The selected representation can be used for classification as input vectors. Since the data structure is maintained as MI-learning, it can be quickly adapted to well-known MI-algorithms based on SVMs without any alteration and with a much smaller size.

Table 1. performance evaluation (Original vs. representation)

Algorithms		Data set1	Data set 2	Data set 3	Data set 4	Data set 5
mi-SVM	Original[1]	0.69±0.15	0.66±0.15	0.58±0.03	0.50±0.05	0.57±0.06
	Adapted[2]	**0.79±0.10**	**0.74±0.15**	**0.75±0.09**	**0.51±0.06**	**0.75±0.08**
MI-SVM	Original	0.77±0.15	0.61±0.08	0.81±0.07	0.50±0.07	0.70±0.10
	Adapted	**0.79±0.12**	**0.76±0.14**	**0.77±0.08**	**0.57±0.13**	**0.76±0.13**
MissSVM	Original	0.69±0.07	0.53±0.09	0.68±0.11	0.46±0.08	0.68±0.12
	Adapted	**0.79±0.13**	**0.76±0.12**	**0.77±0.08**	**0.57±0.13**	**0.76±0.13**
MILES	Original	0.43±0.09	0.61±0.08	0.52±0.03	0.51±0.05	0.51±0.02
	Adapted	0.43±0.09	0.61±0.08	**0.57±0.02**	**0.55±0.03**	**0.51±0.01**
DL-SVM	**Adapted**	0.57±0.09	0.48±0.12	0.49±0.07	0.45±0.07	0.52±0.06

[1]Dataset 1: Musk 1; Dataset 2: Musk 2; Dataset3: Elephant sets; Dataset4: Fox sets; Dataset5: Tiger sests

[2]Adapted indicate the way in the prior study proposed with the input vectors, and with the selected representation.

Table 2. Comparison of task execution time (TETs) (Original vs. representation)

INPUTS	FOLDS	Data set1	Data set2	Data set3	Data set4	Data set5
Original	I	7.56	1753.61	66.12	66.95	46.40
	II	8.87	1826.83	72.53	70.70	49.92
	III	9.13	2375.31	74.30	73.74	46.01
	IV	8.26	1343.86	71.11	68.04	47.28
	V	9.04	1298.29	74.22	65.60	42.49
	VI	8.25	1623.70	73.26	67.27	43.80
	VII	8.02	2517.23	79.02	70.26	46.02
Represen-tative Selection	I	3.77	84.48	10.21	11.50	3.61
	II	3.02	87.91	11.80	6.78	3.72
	III	2.91	87.76	5.25	4.36	3.48
	IV	3.11	66.62	4.55	12.51	9.41
	V	3.23	66.60	10.57	10.25	9.54
	VI	3.13	82.07	10.50	10.46	9.68
	VII	3.05	55.51	10.52	10.07	9.68

[1]Dataset 1: Musk 1; Dataset 2: Musk 2; Dataset3: Elephant sets; Dataset4: Fox sets; Dataset5: Tiger sets

IV. Experiments and Results

We evaluated the performance of the representation on five different benchmark datasets in terms of accuracy using mi-SVM, MI-SVM, MissSVM and MILES, all of which are based on SVMs classifiers for MI-learning. Additionally, we conducted DL-SVM [15]. DL-SVM was proposed to enhance the performance of deep learning by replacing the softmax activation layer at the top level of the architecture with a Linear SVM. Hence, its purpose is to demonstrate a possible implementation within a standard supervised learning framework, rather than solely enhancing performance itself.

Initially, these benchmark algorithms were implemented on the datasets according to the original research description. Subsequently, the selected representation was integrated into these algorithms under the same setting. Each dataset was divided into 7-fold cross-validation with a balanced class distributional setting, and accordingly, 7-fold cross-validation was conducted. The performance results were summarized as the average ±standard deviation in Table 1. The first run was labeled as `original', while the second run, after adapting the representation selection, was labeled as `adapted'. According to Table 1, we observed an improvement in overall accuracies when the proposed selection was adapted to the given MI-algorithms. Additionally, we confirmed that the selection can be performed within the supervised learning framework without any issues.

Computational Efficiency

The computational efficiency was evaluated by task execution time (TET) when the selected representation was integrated into the MI algorithms, compared to the original approach. The task was defined as the procedure from representation selection to prediction. TETs for MissSVM were summarized during 7-fold cross-validation over five independent runs, as MissSVM typically requires higher optimization time. Machine idle time and hardware latency were disregarded in the calculation. We used the CUDA version 12.4 computing platform with an Nvidia GeForce RTX 4090 graphics card, and TETs were measured using the 'time' module in Python 3.9. Table 2 shows the TETs for both the original and the proposed approaches. The average TETs per dataset were 404.77s and 20.90s for the original and proposed approaches, respectively. These results indicate an approximate 83.40% reduction in average TETs when adopting the proposed representation in the algorithm.

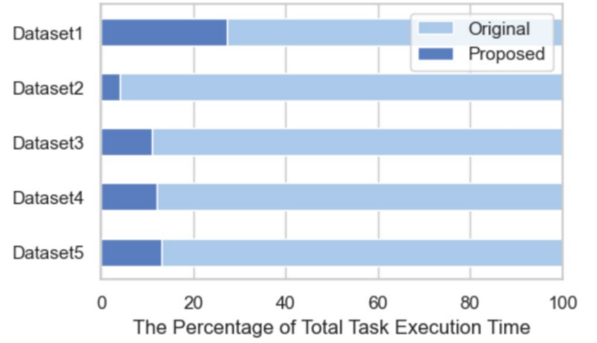

Fig. 2. Relative comparison of TETs.

V. Discussion

SVMs are powerful algorithms in the machine learning domain. In addition to their superior performance, generalization errors are not related to dimensionality [1], [16], which is also an advantage in MI-learning. For this reason, various forms of SVMs have been developed to solve MI problems. Our study proposes an instance selection scheme based on the Hausdorff distance to enhance the performance of SVMs, as the Max-Min operation in the Hausdorff distance can be interpreted similarly to SVMs. Regarding the complex data structure in MI settings, instance selection can offer advantages to SVM classifiers, helping them perform within the standard supervised learning framework. SVM is known to be robust with respect to outliers, especially with a fine-tuned regularization parameter. In particular, when the data are high-dimensional, SVM focuses solely on the support vectors for classification, which helps avoid the risk of estimating a decision boundary that is sensitive to data-specific outliers. These aspects can also be leveraged in our approach.

We evaluated the performance on five benchmark datasets, and confirmed that the adapted representation outperformed the original approach. Additionally, the computational burden was dramatically reduced by more than 80% in terms of average task execution times (TETs). However, there are still some weakness. Since information may not be fully provided, a more aggregated approach could yield better performance depending on the dataset. Moreover, one-to-one mapping to measure distance can be affected by outliers. Nevertheless, it is noteworthy to consider the adaptation of the representation for SVMs due to its computational efficiency and strong performance.

Conflict of Interest

The authors declare no conflict of interest.

Acknowledgment

This research was supported by Basic Science Research Program of the National Research Foundation of Korea (NRF) funded by the Ministry of Education (RS-2023-00237241).

References

[1] Y. Tan and J. Wang, "A support vector machine with a hybrid kernel and minimal Vapnik-Chervonenkis dimension," *IEEE Trans. Knowl. Data Eng.*, vol. 16, pp. 385–395, 2004.

[2] T. Dietterich, R. Lathrop, and T. Lozano-Perez, "Solving the multiple-instance problem with axis-parallel rectangles," *Artif. Intell.*, vol. 89, pp. 31–71, 1997.

[3] P. Viola, J. Platt, and C. Zhang, "Multiple-instance boosting for object detection," in *Proc. Neural Inf. Process. Syst. (NIPS)*, vol. 14, 2006, pp. 1–8.

[4] S. Andrew, I. Tsochantaridis, and T. Hofmann, "Support vector machines for multiple-instance learning," in *Adv. Neural Inf. Process. Syst.*, S. Becker, S. Thrun, and K. Obermayer, Eds. MIT Press, Cambridge, MA, 2003, pp. 561–568.

[5] K. Bennett and A. Demiriz, "Semi-supervised support vector machines," in *Proc. Neural Inf. Process. Syst. (NIPS)*, vol. 11, 1999, pp. 368–374.

[6] Z. Zhou and J. Xu, "On the relation between multi-instance learning and semi-supervised learning," in *Proc. Int. Conf. on Mach. Learn.*, Corvallis, OR, 2007, pp. 1–8.

[7] O. Maron and T. Lozano-Perez, "A framework for multiple-instance learning," in *Adv. Neural Inf. Process. Syst.*, vol. 10, 1997, pp. 570–576.

[8] Q. Zhang and S. Goldman, "EM-DD: an improved multi-instance learning technique," in *Adv. Neural Inf. Process. Syst.*, vol. 14, 2002, pp. 1073–1080.

[9] Z. Zhou and Y. Sun, "Multi-instance learning by treating instances as non-I.I.D samples," in *Proc. Int. Conf. on Mach. Learn. (ICML)*, 2009, pp. 1–18.

[10] Y. Chen, J. Bi, and J. Wang, "MILES: Multiple-instance learning via embedded instance selection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, pp. 1931–1947, 2006.

[11] H. Zhang and D. Tao, "Deep multiple instance learning with distance-aware self-attention," *arXiv preprint arXiv:2305.10552*, 2023.

[12] M. Liu, "The semiproximal SVM approach for multiple instance learning: A kernel-based computational study," *Electronics*, vol. 10, no. 20, art. 2459, 2024, doi: 10.3390/electronics10202459.

[13] A. Tuzhilin, "Who invented the gromov-hausdorff distance?" *arXiv preprint arXiv:1612.00728v1*, 2016, pp. 1–7.

[14] E. Rodrigues, "An efficient and locality-oriented hausdorff distance algorithm: Proposal and analysis of paradigms and implementations," *Pattern Recognit.*, vol. 117, pp. 1–11, 2021.

[15] Y. Tang, "Deep learning using support vector machines," *arXiv preprint arXiv:1306.0239*, 2013, pp. 1–5.

[16] M. Hearst, "Support vector machines," *IEEE Intell. Syst.*, vol. 18, pp. 18–28, 1998.