Attention-to-Embedding Framework for Multi-instance Learning

Mei Yang[1], Yu-Xuan Zhang[1], Mao Ye[3], and Fan Min[1,2(✉)]

[1] School of Computer Science, Southwest Petroleum University, Chengdu, China
{yangmei,minfan}@swpu.edu.cn
[2] Institute for Artificial Intelligence, Southwest Petroleum University, Chengdu, China
[3] School of Computer Science and Engineering, University of Electronic Science and Technology, Chengdu, China

Abstract. We present an attention-to-embedding framework that explicitly addresses the challenge posed by multi-instance learning (MIL) classification tasks, where learning objects are bags containing various numbers of instances. Two key issues of this work are to extract relevant information by determining the relationship between the bag and its instances, and to embed the bag into a new feature space. To respond to these problems, a network with the popular attention mechanism is designed that assigns a new representation and a class probability vector to a given instance in the bag. In addition, compared with the traditional MIL methods, we offer a new embedding function according to the assigned results of instances to process the bag embedding that is unrelated to the distance metric. As a result, MIL challenges will be reduced to single-instance learning (SIL) problems that can be solved using basic machine learning algorithms such as SVM. Extensive experiments on thirty-four data sets demonstrate that our proposed method has the best overall performance over other state-of-the-art MIL methods. This strategy, in particular, has a substantial advantage on web data sets and better stability. *Source codes are available at* https://github.com/InkiInki/AEMI.

Keywords: Attention · Embedding · Multi-instance learning · Network

1 Introduction

Multi-instance learning (MIL) was originally designed for drug activity prediction [4]. In contrast to traditional single-instance learning (SIL), each object in MIL is a bag containing various numbers of instances. A label is assigned to the bag, but not to the individual instances. To date, MIL has also been frequently utilized in a variety of applications, such as image classification [15], text categorization [14], sentiment analysis [1], and web index recommendation [10].

J. Gama et al. (Eds.): PAKDD 2022, LNAI 13281, pp. 109–121, 2022.
https://doi.org/10.1007/978-3-031-05936-0_9

Over the years, many excellent algorithms for MIL classification tasks have been proposed. Traditional MIL covers but is not limited to the following solutions: a) Instance-based approaches calculate the bag label by predicting the instance label and combining MIL assumptions [6]; b) Bag-based approaches treat each bag as an atom and train a classifier based on bag-level metrics, such as graph kernel [19] and isolation set-kernel [14]; and c) Embedding-based approaches transform bags into a new feature space and establish the learning process with SIL methods [13]. Neural network-based MIL can be categorized into two types [11]: a) mi-Net uses an instance-level classifier to obtain the instance probabilities. As a result, the bag label is derived using instance probabilities and the convex max operator (or max operator); and b) MI-net builds a fixed-length vector as the new representation of the bag and learns a bag-level classifier directly to obtain the bag label.

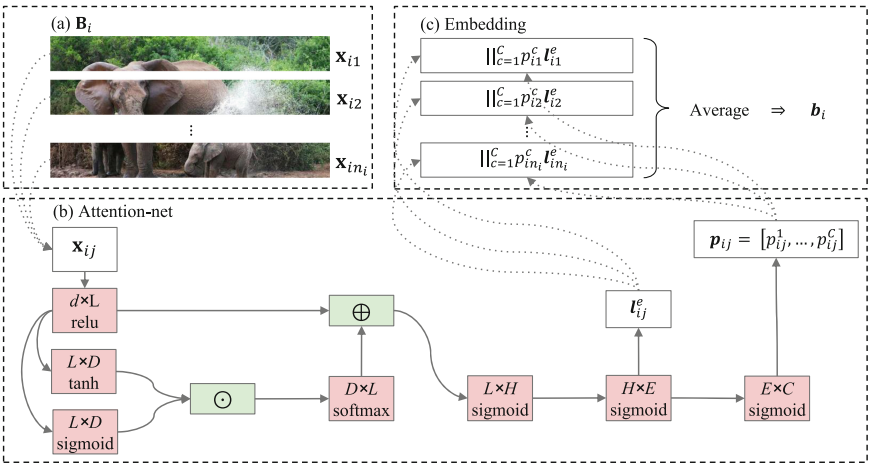

Fig. 1. The overall framework of AEMI: a) The original bag $\mathbf{B}_i$ with a series of unlabeled instances $\mathbf{x}_{ij}$; b) The attention-net for extracting instance information; and c) The bag-level embedding is used to transform each bag into a new feature space. In addition, d is the dimension; L, D, H and E are the number of nodes; C is the number of classes; $\odot$ and $\oplus$ are element-wise multiplication and addition, respectively; $\boldsymbol{l}_{ij}^{e}$ and $\boldsymbol{p}_{ij}$ are the new representation and class probability vector of $\mathbf{x}_{ij}$, respectively.

In this paper, we propose a new attention-to-embedding framework (AEMI) to handle multi-instance learning classification tasks. Figure 1 shows the AEMI's overall framework, which innovatively combines the attention mechanism derived from neural networks and the MIL embedding method. The first part is a sample image that can be regarded as a bag, with each sub-area corresponding to one of the bag's instances. This part reflects the challenge that the neural network must face when applied to MIL: The label of the image is known, but the label of the instance contained within it is unknown. In the previous MIL neural

network-based approaches, the relationship between instances and the bag is commonly determined via pooling functions or attention coefficients. While in our configuration, the embedding function needs to ensure that the embedded bags can be distinguished.

Therefore, we provide an attention network whose input is the instance and outputs are its new representation and class probability vector. By designing an embedding function and controlling the size of the representation, each instance can be embedded as a vector, and each bag can be transformed into the same space by the arithmetic average of all embedded vectors, as shown in the third part. As a result, any traditional SIL classifier can be employed to train a model.

The contributions of this work are summarized as follows:

- We convert MIL tasks into SIL ones by designing a framework that connects MIL neural networks with the embedding method. This has the advantage of alleviating the issue of neural network classification instability caused by random parameter initialization.
- We design a network that is not reliant on the MIL pooling function and an embedding function without distance metrics. Specifically, the class probability vector of each instance in the bag is introduced into the embedding process to improve the distinguishability of embedding bags.

Experiments are undertaken on thirty-four MIL classification data sets to quantify the performance of AEMI. These data sets come from a variety of fields, including drug activity prediction, text classification, image classification tasks and web index recommendation tasks. In most cases, the experimental results show that AEMI outperforms state-of-the-art algorithms and has demonstrated significant benefits on web data sets.

2 Related Work

In this section, we will briefly introduce the related work, including MIL attention neural networks and embedding methods.

2.1 MIL Attention Neural Networks

The attention mechanism is commonly employed in deep learning for text analysis [2] or image recognition [5]. However, few studies have focused on the attention mechanism of MIL. Attention-net [8] incorporates interpretability into the MIL method and increases its flexibility. Loss-attention [9] connects the attention mechanism with the loss function. Unlike prior techniques, we exclusively employ neural networks to obtain the new representations and class probability vector for each instance in the bag. Therefore, the network we designed does not depend on the MIL pooling function, and its input is the instance assigned as the corresponding bag label.

2.2 MIL Embedding Methods

Multi-instance embedding methods' core idea is to transform the bag into a new feature space and train a model with traditional machine learning methods. miFV [12] extracts information from the instance space using the Gaussian mixture model and derives the embedded vector using the Fisher vector, while the time complexity of this technique increases as the dimensionality of the data set grows. MILDM [13] designs an instance evaluation function to select instances with the most discriminativeness and builds a mapping pool to embed bags. StableMIL [17] builds upon identifying a novel connection between MIL and the potential outcome framework in causal effect estimation. The majority of these methods rely on distance metrics, such as the bag-level average Hausdorff distance [16] and the bag-instance minimum distance [13]. However, the adaptability of different distance measures to different types of data sets may be completely different. Therefore, we design an embedding function by combining the instance's new representation and probability vector derived from the attention-net.

3 Methodology

In this section, we will first describe the preliminaries of the proposed AEMI algorithm. Then, the attention-net and bag-level embedding are introduced as part of AEMI. Finally, we have some discussions about this method.

Let $\mathcal{B} = \{\mathbf{B}_i\}_{i=1}^{N}$ be a MIL data set with N bags, where $\mathbf{B}_i = \{\mathbf{x}_{ij}\}_{j=1}^{n_i} \in \mathcal{B}$ is a bag with n_i instances, $\mathbf{x}_{ij} \in \mathbf{B}_i$, $n_i = |\mathbf{B}_i|$ and d is the dimension. Let $\mathbf{Y} = [y_1, \ldots, y_N]$ be the label vector corresponding $\mathcal{B}$, where $y_i \in \{1, \ldots, C\}$ is the label of $\mathbf{B}_i$ and C is the number of classes. With the basic MIL assumption [4], label y_i is supposed that: a) $\mathbf{B}_i$ is labeled as c-th class iff it contains at least one c-th class instance; and b) $\mathbf{B}_i$ contains an instance belonging to two or more classes is impossible.

Our goal is to transform the MIL tasks into SIL ones by connecting the attention-net with the bag-level embedding.

3.1 The Attention-Net

The core of MIL attention network [8] is to calculate an attention coefficient for each instance $\mathbf{x}_{ij}$:

$$\alpha_{ij} = \boldsymbol{w}^{\mathrm{T}}(\tanh(\boldsymbol{V}\boldsymbol{p}_{ij}^{\mathrm{T}}) \odot \mathrm{sigmoid}(\boldsymbol{U}\boldsymbol{p}_{ij}^{\mathrm{T}})), \tag{1}$$

where $\boldsymbol{w} \in \mathbb{R}^{L\times 1}$ and $\boldsymbol{V}, \boldsymbol{U} \in \mathbb{R}^{L\times C}$ are parameters of neural network, L is the number of fully connected layer's nodes, C is the number of classes, $\boldsymbol{p}_{ij}$ is the class probability vector of the instance $\boldsymbol{x}_{ij}$ and $\odot$ is element-wise multiplication.

To extract the information of $\boldsymbol{x}_{ij}$ and construct the embedding function, we modify this mechanism to generate the attentional representation of $\mathbf{x}_{ij} \in \mathbf{B}_i$

$$\boldsymbol{h}_{ij}^{a} = softmax((\tanh(\boldsymbol{h}_{ij}^{r}\boldsymbol{W}^{t}) \odot \mathrm{sigmoid}(\boldsymbol{h}_{ij}^{r}\boldsymbol{W}^{s}))\boldsymbol{W}^{o}), \tag{2}$$

where

$$\boldsymbol{h}_{ij}^{r} = \text{relu}(\mathbf{x}_{ij}\boldsymbol{W}^{r}), \tag{3}$$

and $\boldsymbol{W}^{t}, \boldsymbol{W}^{s} \in \mathbb{R}^{L\times D}$, $\boldsymbol{W}^{o} \in \mathbb{R}^{L\times D}$ and $\boldsymbol{W}^{r} \in \mathbb{R}^{d\times L}$ are parameters of the designed attention-net $f_{\psi}(\cdot)$, D is the number of nodes and d is the dimension. In addition, $\boldsymbol{h}_{ij}^{a}$ and $\boldsymbol{h}_{ij}^{r}$ are merged into

$$\boldsymbol{h}_{ij}^{m} = \boldsymbol{h}_{ij}^{a} \oplus \boldsymbol{h}_{ij}^{r}, \tag{4}$$

where $\oplus$ is element-wise addition. To improve the information extraction capabilities of the network, and get the new representation and class probability vector of instance, we add the following fully connected layers:

$$\begin{aligned} \boldsymbol{l}_{ij}^{h} &= \text{sigmoid}(\boldsymbol{h}_{ij}^{m}\boldsymbol{W}^{h}), \\ \boldsymbol{l}_{ij}^{e} &= \text{sigmoid}(\boldsymbol{l}_{ij}^{h}\boldsymbol{W}^{e}), \\ \boldsymbol{p}_{ij} &= \text{sigmoid}(\boldsymbol{l}_{ij}^{e}\boldsymbol{W}^{c}), \end{aligned} \tag{5}$$

where $\boldsymbol{W}^{h} \in \mathbb{R}^{L\times H}$, $\boldsymbol{W}^{e} \in \mathbb{R}^{H\times E}$, $\boldsymbol{W}^{c} \in \mathbb{R}^{H\times C}$, H and E are the number of nodes.

Network $f_{\psi}(\cdot)$ will participate in the construction of the embedding function, and the most basic requirement it needs to meet is to determine the class of $\mathbf{x}_{ij}$. Therefore, we set the input of $f_{\psi}(\cdot)$ to $\mathbf{x}_{ij}$, and its label will be assigned as y_i. The benefits include the following: a) Instance label and bag label to ensure uniformity; b) The training process is not affected by the bag structure; and c) With appropriate modifications, most existing MIL neural networks can be applied. Finally, we define the loss function as

$$\ell = -\sum_{i=1}^{N}\sum_{j=1}^{n_i} \log \frac{\exp(p_{ij}^{y_i})}{\sum_{c}\exp(p_{ij}^{c})}, \tag{6}$$

where $p_{ij}^{c} \in \boldsymbol{p}_{ij}$.

3.2 The Bag-Level Embedding

The embedding function is used to transform a bag into a new feature space, and its general definition is as follows:

$$\mathcal{F}^{B}(\mathbf{B}_i) \mapsto \mathbf{B}_i = [d(\mathbf{B}_1, \boldsymbol{K}_1), \ldots, d(\mathbf{B}_{|\mathcal{K}|}, \boldsymbol{K}_{|\mathcal{K}|})], \tag{7}$$

where $\mathcal{K}$ is a key sample set derived from the bag space $\mathcal{B}$, K_i is the i-th sample of $\mathcal{K}$, and $d(\cdot,\cdot)$ is the distance between the bag and the key sample. By specifying the size of $\mathcal{K}$, the bag $\mathbf{B}_i$ can be embedded as a vector $\mathbf{b}_i$ in the new feature space.

One disadvantage of Eq. (7) is that the employed $d(\cdot,\cdot)$ has a significant impact on embedding results. Therefore, by considering the probability distribution of

instances in the bag, we design a new bag-level embedding without distance metrics as

$$\mathcal{F}^N(\mathbf{B}_i) \mapsto \mathbf{b}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} \mathcal{F}^I(\mathbf{x}_{ij}), \tag{8}$$

where

$$\mathcal{F}^I(\mathbf{x}_{ij}) = \|_{c=1}^{C} p_{ij}^c \boldsymbol{l}_{ij}^e, \tag{9}$$

where $\|$ denotes concatenation. For example, for an input instance $\mathbf{x}_{ij}$, we assume that the corresponding outputs are $\boldsymbol{p}_{ij} = [0.8, 0.2]$ and $\boldsymbol{l}_{ij}^e = [0.4, 0.5]$. As a result, $\mathcal{F}^I(\mathbf{x}_{ij}) = [0.32, 0.4] \| [0.08, 0.1] = [0.32, 0.4, 0.08, 0.1]$. The advantage of this strategy is that the difference between the embedding results of the two instances will be positively correlated with their class probability vectors.

Algorithm 1 presents the pseudo code of the AEMI algorithm. Line 1 generates the instance space $\mathcal{X}$ by collecting all instances of the data set $\mathcal{B}$, and uses it as the input to the attention-net $f_\psi(\cdot)$. Line 2 assigns the label of instance $\mathbf{x}_{ij}$ as $y_i \in \mathbf{Y}$ with the goal of allowing $f_\psi(\cdot)$ to distinguish as accurately as possible the instances in different classes of bags. Line 3 generates a single-instance label vector $\mathcal{L}$ and uses it to participate in the loss calculation. Line 4 trains $f_\psi(\cdot)$ with these generations. Lines 5–9 embed each bag $\mathbf{B}_i$ into $\boldsymbol{b}_i$ according to the designed embedding function, and merge it into $\mathbf{X}$.

Based on the set of embedding vector $\mathbf{X}$ and its corresponding label vector $\mathbf{Y}$, we can train a classification model $\mathcal{M}$ with a traditional single-instance classifier.

Algorithm 1. The AEMI algorithm

Input:
Data set $\mathcal{B}$;
Label vector $\mathbf{Y}$;
Output:
Single-instance classifier $\mathcal{M}$;
The trained neural network $f_\psi(\cdot)$;
1: $\mathcal{X} = \{\mathbf{x}_{ij} | i \in [1..N], j \in [1..n_i]\}$, where $\mathbf{x}_{ij} \in \mathbf{B}_i \in \mathcal{B}$, $N = |\mathcal{B}|$ and $n_i = |\mathbf{B}_i|$;
2: Assign the label of instance $\mathbf{x}_{ij}$ as y_i, where $y_i \in Y$ is the label of $\mathbf{B}_i$;
3: Generate the single-instance label vector $\mathcal{L}$ by collecting all the assigned instance labels;
4: Train a neural network $f_\psi(\cdot)$ with $\mathcal{X}$ and $\mathcal{L}$;
5: $\mathbf{X} = \emptyset$;
6: **for** $(i \in [1..N])$ **do**
7: Embed bag $\mathbf{B}_i$ into $\mathbf{b}_i$ according to Eq. (8) with $f_\psi(\cdot)$;
8: $\mathbf{X} \leftarrow \mathbf{X} \cup \{\mathbf{b}_i\}$;
9: **end for**
10: Train a single-instance classifier $\mathcal{M}$ with $\mathbf{X}$ and $\mathbf{Y}$;
11: Output $\mathcal{M}$ and $f_\psi(\cdot)$;

Proposition 1. *The time complexity of Algorithm 1 is $O(\varepsilon dn)$, where ε, d, and n are the number of epochs, dimensions, and total instances in all bags, respectively.*

Proof. Let the number of bags be N. The instance space $\mathcal{X}$ and its corresponding label vector $\mathcal{L}$ are generated in Lines 1–3. Their time complexity is $O(n)$. In Line 4, the training of neural network costs $O(\varepsilon dn)$. Lines 5–9 embed each bag into a new feature space, which costs $O(dn)$. Line 10 trains a single-instance classifier, which costs $O(EN)$. Generally, we have $E < d$ and $N \ll n$. Therefore, the total time complexity of AEMI is $O(\varepsilon dn)$.

3.3 Scheme Analysis

The following are two characteristics of the designed attention-net: a) Adaptability: The structure of this attention-net is adaptively adjusted according to the size of the data set, i.e., any dimensional instance can be represented in a vector. To put it another way, this network can function normally with the default parameter settings; and b) Interpretability: Ideally, $\forall \mathbf{x}_{ij} \in \mathbf{B}_i, y_i = c, p_{ij}^c \geq p_{ij}^k$, where $k \in [1..C]$. Therefore, our goal is to construct an embedding function with higher distinguishability by making the training results of the designed network fit this state as much as possible.

By combining embedding-based approaches with neural networks, the AEMI algorithm is designed to transform MIL tasks to the SIL ones. With this algorithm, each bag $\mathbf{B}_i$ can be embedded as a vector $\mathbf{b}_i \in \mathbb{R}^{CE}$ in the new feature space. According to embedding results, we may encounter such a dilemma. When $CE \geq d$, where d is the dimension, the increased dimensionality of the embedding vector may cause some noise.

4 Experiments

In this section, we will firstly describe the used data sets and the comparison algorithms. Then, the AEMI algorithm is put to the test in comparison against seven state-of-the-art approaches in a series of experiments.

4.1 Data Sets

We conducted experiments on four types of MIL data sets: Drug activity prediction, text classification, image classification data sets, and web index recommendation data sets. All of these data sets can be found at https://blog.csdn.net/weixin_44575152/article/details/104769348.

Drug Activity Prediction. The benchmark data sets musk1 and musk2 are commonly used in drug activity prediction tasks [4]. Its goal is to predict whether a new molecule can be used to make a drug. In the MIL domain, a musk molecule is represented as a bag with a variable number of 166-dimensional instances. According to the basic MIL assumption, a molecule is positive iff it possesses at least one instance that can be used to make a drug; otherwise negative.

Text Categorization. To conduct experiments, we employed ten text data sets derived from the Newsgroups corpus. Each data set contains 50 positive and 50 negative bags. Each positive bag contains 3% of posts from the specified positive class and the rest from other classes, whereas instances of negative bags are randomly drawn from the non-main class. Each instance is also represented by a 200-dimensional TFIDF feature.

Image Classification. Corel with 100 categories is a famous database for the image classification task [3]. Each category contains 100 images in JPG format with a shape of 187×126 or 126×187. Elephant and tiger are from the Corel database, and all of them have been preprocessed by the Blobworld bag generator. To consider a more challenging scenario, we built ten mnist-bag data sets with the mnist classification data set. Take mnist0 as an example. The generation details are as follows: a) Set the number of positive and other class bags to 100; b) Set the minimum and maximum size of bags to 10 and 50, respectively; c) Set the minimum and maximum number of the positive instances in the positive bag to 2 and 8, respectively; d) Each positive instance is an image randomly selected from the 0-th class of the mnist data set, while the other instance is from the other classes; and e) The selected image will be stretched as a 786-dimensional instance. The random seed of the generating algorithm will be fixed for experimental fairness.

Web Index Recommendation. The purpose of web index recommendation is to recommend interesting web page indexes to particular users. Each of the nine sub data sets in the web data set corresponds to a user's evaluation of a web page [18]. Each web page serves as a bag and links on the page serves as instances. Since web page processing is connected to word frequency, web data sets have high-dimensionality and sparsity.

4.2 Comparative Algorithms

As a comparison, we employed seven start-of-the-art MIL classification algorithms listed below: a) BAMIC [16] and miVLAD [12] use the clustered centers of bag- and instance-level kMeans as key samples, respectively; b) miFV [12] uses the Gaussian mixture model to extract information of the data set; c) MILDM [13] selects the key samples with the discriminative instance evaluation criterion; d) MILFM [7] treats all instances of positive bags and the clustered centers of other bags as key samples; and e) Attention-net [8] and loss-attention [9] are two popular MIL networks. Table 1 shows the parameter settings for AEMI and the above algorithms.

Table 1. Parameter settings.

Algorithm	Parameter	Setting
AEMI	Epoch for musk, elephant and tiger	100
	Epoch for others	5
	Learning rate	0.001
	Number of nodes E	Number of bags N
BAMIC	Number of clustering centers	N
	Distance metric	Average Hausdorff with Euclidean distance [16]
miFV	Components of Gaussian mixture model	1
miVLAD	Size of Code book	1
MILDM	Distance metric	Bag-instance maximum distance [13] with gamma 1
	Instance selection mode	Global
	Number of discriminative instances	N
MILFM	Number of clustering centers	50
	Distance metric	Same as MILDM
Attention-net	Epoch and Learning rate	Same as AEMI
Loss-attention	Epoch	Same as AEMI
	Learning rate	0.0001

4.3 Experimental Results

Tables 2 shows the experimental results of the AEMI and seven rival algorithms based on three SIL classifier kNN, SVM and J48. The best accuracy value for each data set is highlighted with "•". Average ($d < 1000/d \geq 1000$) denotes the average classification performance across data sets in the specified dimension range. The results demonstrate that the AEMI algorithm has a significant advantage on web recommendation and the mnist data sets. Specifically, the accuracy of AEMI is about 10% greater than that of competing methods on some data sets, such as mnist9 and web4, and the average ones are 3.6% than in second place and 30.4% than in penultimate place when $d < 1000$. The following reasons may apply: a) The proposed attention-net can effectively extract information from web data sets and generate the new representation and class probability vector for instances; and b) The embedding mechanism converts the bag into the new feature space while preserving as much information as possible.

Furthermore, some results necessitate further care. a) On the text categorization data sets, AEMI achieves a moderate outcome, while BAMIC, miFV, and miVLAD get relatively large advantages. For example, miFV has a considerable edge on the news.mf data set. The reason for this could be that the Gaussian mixture model of miFV can effectively mine the information of this type of data sets. While the news.mf's embedding results of AEMI may contain some noise; and b) MILDM and MILFM have inadequate impacts on text and web data sets. All three methods find key instances in the specified instance space. Take the "flower"/"other" images as an example, the number of "flower"-instances is usually less than the number of "other"-instances. As a result, these selected key instances may not be "key".

Table 2. Performance comparison between AEMI and rival algorithms. Experiments were run 5 times 10CV and an average of the classification accuracy (± the standard deviation) is reported.

Data set	(d)	BAMIC	miFV	miVLAD	MILDM	MILFM	Attention-net	Loss-attention	**AEMI**
Musk1	(166)	0.891 ± 0.011	$0.920 \pm 0.008\bullet$	0.847 ± 0.011	0.824 ± 0.025	0.871 ± 0.005	0.884 ± 0.022	0.890 ± 0.020	0.867 ± 0.019
Musk2	(166)	$0.860 \pm 0.011\bullet$	0.848 ± 0.015	0.780 ± 0.054	0.826 ± 0.016	0.822 ± 0.035	0.822 ± 0.047	0.848 ± 0.019	0.804 ± 0.010
News.aa	(200)	0.852 ± 0.010	0.834 ± 0.016	0.836 ± 0.027	0.510 ± 0.050	0.510 ± 0.000	0.862 ± 0.019	$0.874 \pm 0.016\bullet$	0.808 ± 0.023
News.cg	(200)	$0.812 \pm 0.004\bullet$	0.802 ± 0.008	0.790 ± 0.014	0.526 ± 0.052	0.504 ± 0.010	0.610 ± 0.017	0.644 ± 0.033	0.782 ± 0.016
News.mf	(200)	0.696 ± 0.019	$0.736 \pm 0.016\bullet$	0.716 ± 0.029	0.488 ± 0.037	0.510 ± 0.006	0.666 ± 0.022	0.716 ± 0.032	0.676 ± 0.037
News.rm	(200)	0.808 ± 0.016	$0.877 \pm 0.020\bullet$	0.812 ± 0.016	0.546 ± 0.041	0.530 ± 0.026	0.854 ± 0.021	0.871 ± 0.026	0.818 ± 0.013
News.rsh	(200)	0.828 ± 0.010	0.884 ± 0.010	0.894 ± 0.010	0.442 ± 0.021	0.500 ± 0.000	0.872 ± 0.005	$0.914 \pm 0.012\bullet$	0.884 ± 0.022
News.sc	(200)	0.774 ± 0.010	0.750 ± 0.018	$0.818 \pm 0.023\bullet$	0.518 ± 0.042	0.512 ± 0.004	0.780 ± 0.014	0.802 ± 0.036	0.800 ± 0.026
News.se	(200)	$0.938 \pm 0.004\bullet$	0.926 ± 0.005	0.918 ± 0.008	0.574 ± 0.061	0.530 ± 0.000	0.554 ± 0.010	0.572 ± 0.036	0.864 ± 0.014
News.tpmd	(200)	0.830 ± 0.000	0.799 ± 0.016	0.832 ± 0.015	0.554 ± 0.019	0.554 ± 0.048	0.836 ± 0.016	$0.844 \pm 0.012\bullet$	0.788 ± 0.033
News.tpmi	(200)	0.690 ± 0.011	0.752 ± 0.015	$0.766 \pm 0.015\bullet$	0.482 ± 0.037	0.506 ± 0.005	0.720 ± 0.013	0.482 ± 0.022	0.710 ± 0.011
News.trm	(200)	0.728 ± 0.008	0.740 ± 0.014	$0.786 \pm 0.022\bullet$	0.466 ± 0.048	0.510 ± 0.011	0.606 ± 0.060	0.514 ± 0.064	0.720 ± 0.026
Elephant	(230)	0.762 ± 0.012	0.852 ± 0.013	0.853 ± 0.010	0.765 ± 0.022	0.817 ± 0.023	0.848 ± 0.014	0.872 ± 0.005	$0.875 \pm 0.010\bullet$
Tiger	(230)	0.704 ± 0.011	0.789 ± 0.006	$0.843 \pm 0.008\bullet$	0.692 ± 0.008	0.754 ± 0.006	0.810 ± 0.031	0.819 ± 0.011	0.814 ± 0.009
Mnist0	(786)	0.913 ± 0.018	0.820 ± 0.009	0.873 ± 0.002	0.484 ± 0.015	0.507 ± 0.002	0.979 ± 0.005	$0.995 \pm 0.003\bullet$	0.985 ± 0.010
Mnist1	(786)	0.978 ± 0.004	0.724 ± 0.013	0.845 ± 0.013	0.803 ± 0.012	0.975 ± 0.006	0.873 ± 0.146	$0.992 \pm 0.004\bullet$	0.980 ± 0.003
Mnist2	(786)	0.773 ± 0.021	0.858 ± 0.009	0.910 ± 0.008	0.462 ± 0.008	0.496 ± 0.028	0.959 ± 0.019	0.966 ± 0.005	$0.973 \pm 0.005\bullet$
Mnist3	(786)	0.865 ± 0.015	0.787 ± 0.005	0.863 ± 0.004	0.556 ± 0.009	0.580 ± 0.024	0.940 ± 0.006	0.942 ± 0.019	$0.956 \pm 0.006\bullet$
Mnist4	(786)	0.855 ± 0.006	0.757 ± 0.011	0.810 ± 0.017	0.451 ± 0.012	0.520 ± 0.034	0.931 ± 0.014	0.896 ± 0.024	$0.937 \pm 0.011\bullet$
Mnist5	(786)	0.759 ± 0.023	0.759 ± 0.016	0.831 ± 0.009	0.487 ± 0.028	0.496 ± 0.008	0.922 ± 0.020	0.838 ± 0.039	$0.964 \pm 0.006\bullet$
Mnist6	(786)	0.914 ± 0.006	0.837 ± 0.017	0.852 ± 0.007	0.466 ± 0.026	0.460 ± 0.034	0.927 ± 0.037	$0.963 \pm 0.007\bullet$	0.959 ± 0.006
Mnist7	(786)	0.908 ± 0.012	0.859 ± 0.013	0.855 ± 0.006	0.530 ± 0.027	0.629 ± 0.037	$0.986 \pm 0.004\bullet$	0.974 ± 0.004	0.975 ± 0.008
Mnist8	(786)	0.786 ± 0.035	0.731 ± 0.020	0.808 ± 0.005	0.494 ± 0.021	0.507 ± 0.002	0.879 ± 0.035	0.749 ± 0.062	$0.926 \pm 0.014\bullet$
Mnist9	(786)	0.837 ± 0.017	0.742 ± 0.017	0.797 ± 0.005	0.583 ± 0.023	0.516 ± 0.017	0.845 ± 0.010	0.771 ± 0.048	$0.958 \pm 0.005\bullet$
Web1	(5863)	$0.844 \pm 0.016\bullet$	0.838 ± 0.007	0.813 ± 0.018	0.838 ± 0.007	0.824 ± 0.012	0.811 ± 0.015	0.811 ± 0.140	0.809 ± 0.013
Web2	(6519)	0.806 ± 0.024	0.826 ± 0.007	0.818 ± 0.013	0.833 ± 0.009	0.820 ± 0.023	0.807 ± 0.019	0.820 ± 0.006	$0.838 \pm 0.018\bullet$
Web3	(6306)	0.815 ± 0.024	0.826 ± 0.009	$0.827 \pm 0.012\bullet$	0.826 ± 0.007	0.815 ± 0.020	0.813 ± 0.009	0.813 ± 0.007	0.813 ± 0.023
Web4	(6059)	0.765 ± 0.004	0.807 ± 0.012	0.844 ± 0.015	0.806 ± 0.015	0.804 ± 0.020	0.844 ± 0.027	0.785 ± 0.009	$0.916 \pm 0.011\bullet$
Web5	(6407)	0.789 ± 0.004	0.782 ± 0.061	0.822 ± 0.014	0.787 ± 0.021	0.781 ± 0.011	0.822 ± 0.015	0.776 ± 0.011	$0.895 \pm 0.009\bullet$
Web6	(6417)	0.809 ± 0.019	0.778 ± 0.005	0.847 ± 0.016	0.846 ± 0.021	0.816 ± 0.022	0.811 ± 0.020	0.782 ± 0.005	$0.920 \pm 0.012\bullet$
Web7	(6450)	0.558 ± 0.016	0.687 ± 0.030	0.742 ± 0.012	0.602 ± 0.037	0.566 ± 0.018	0.713 ± 0.021	0.485 ± 0.031	$0.786 \pm 0.019\bullet$
Web8	(5999)	0.504 ± 0.032	0.706 ± 0.021	0.727 ± 0.021	0.544 ± 0.016	0.576 ± 0.032	0.713 ± 0.012	0.466 ± 0.050	$0.806 \pm 0.024\bullet$
Web9	(6279)	0.500 ± 0.015	0.753 ± 0.022	0.758 ± 0.021	0.551 ± 0.021	0.591 ± 0.021	0.724 ± 0.039	0.503 ± 0.021	$0.809 \pm 0.026\bullet$
Average ($d < 1000$)		0.823 ± 0.012	0.808 ± 0.013	0.831 ± 0.014	0.564 ± 0.028	0.588 ± 0.015	0.832 ± 0.025	0.823 ± 0.023	$0.868 \pm 0.014\bullet$
Average ($d \geq 1000$)		0.710 ± 0.017	0.778 ± 0.019	0.800 ± 0.016	0.737 ± 0.017	0.733 ± 0.020	0.784 ± 0.020	0.693 ± 0.031	$0.844 \pm 0.017\bullet$

Table 3 shows the comparison results of the maximum and minimum classification accuracy of the AEMI algorithm and an attention network method. The terms "net" represents the gate-attention network [8] used for comparison and "our" denotes specifically to the comparison of AEMI's SVM classification results. The symbol $\diamond/\star$ means that the difference between the maximum value minus the minimum value is greater than or equal to 0.05/0.1. The results show that AEMI can alleviate the instability of the neural network caused by parameter initialization without reducing the classification performance. In the mnist2 data set, for example, the net method's accuracy varies by 37.5%, while ours varies by only 1%.

Table 3. Performance comparison between AEMI and gate-attention network. Experiments were run 5 times 10CV and minimum/maximum of the classification accuracy (± the standard deviation) is reported.

Data set	Net (Min)	Net (Max)	Our (Min)	Our (Max)
Mnist0	⋄0.915	⋄0.970	0.970	0.995
Mnist1	⋆0.585	⋆0.960	0.975	0.985
Mnist2	0.940	0.950	0.965	0.980
Mnist3	0.935	0.980	0.955	0.965
Mnist4	⋄0.895	⋄0.945	0.920	0.950
Mnist5	⋆0.860	⋆0.960	0.955	0.970
Mnist6	⋆0.775	⋆0.960	0.950	0.965
Mnist7	0.975	0.990	0.965	0.985
Mnist8	0.875	0.915	0.910	0.945
Mnist9	⋄0.820	⋄0.870	0.950	0.960
Web1	0.800	0.846	0.791	0.827
Web2	0.773	0.818	⋄0.809	⋄0.864
Web3	0.800	0.836	⋄0.791	⋄0.855
Web4	⋄0.818	⋄0.873	0.900	0.927
Web5	⋄0.773	⋄0.855	0.882	0.900
Web6	⋄0.791	⋄0.855	0.900	0.927
Web7	0.700	0.746	0.764	0.809
Web8	0.709	0.745	⋄0.764	⋄0.836
Web9	⋄0.682	⋄0.736	⋄0.773	⋄0.846

5 Conclusion and Further Work

We propose the AEMI algorithm to train an attention-net based on the relationship between the bag and its instances, and use an embedding function to transform MIL tasks into SIL ones. The experimental results of studies prove that AEMI is superior to state-of-the-art MIL classification methods, has significant advantages, especially on web data sets, and has relatively stable classification performance. In addition, the majority of the rival MIL methods perform poorly on MIL web recommendation and mnist, and the neural network-based methods' outcomes of successive experiments may be substantially different due to the random setting of the parameter initialization.

The following topics deserve further investigation:

– More flexible embedding functions. Web data sets with thousands of features can be effectively reduced in dimensionality using the proposed embedding function. However, this may increase the dimensionality of these relatively low-dimensional data sets after embedding. As a result, these may be a factor in their moderate performance on some data sets, such as musk1 and tiger.

– More efficient neural networks. On most data sets, the designed attention-net only requires 5 epochs of training to achieve good results, but on few data sets like musk1, it requires more epochs. Some details are shown in Table 1.

Acknowledgements. This work was supported in part by the National Key R&D Program of China (2018YFE0203900), National Natural Science Foundation of China (61773093), Sichuan Science and Technology Program (2020YFG0476), Important Science and Technology Innovation Projects in Chengdu (2018-YF08-00039-GX), and Central Government Funds of Guiding Local Scientific and Technological Development (2021ZYD0003).

References

1. Angelidis, S., Lapata, M.: Multiple instance learning networks for fine-grained sentiment analysis. Trans. Assoc. Comput. Linguist. **6**, 17–31 (2018)
2. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate (2014)
3. Chen, Y.X., Bi, J.B., Wang, J.Z.: MILES: multiple-instance learning via embedded instance selection. IEEE Trans. Pattern Anal. Mach. Intell. **28**(12), 1931–1947 (2006)
4. Dietterich, T.G., Lathrop, R.H., Lozano-Pérez, T.: Solving the multiple instance problem with axis-parallel rectangles. Artif. Intell. **89**(1–2), 31–71 (1997)
5. Fu, J.L., Zheng, H.L., Mei, T.: Look closer to see better: recurrent attention convolutional neural network for fine-grained image recognition. In: Computer Vision and Pattern Recognition, pp. 4438–4446 (2017)
6. He, C.K., Shao, J., Zhang, J.S., Zhou, X.M.: Clustering-based multiple instance learning with multi-view feature. Expert Syst. Appl. **162**, 113027 (2020)
7. Hong, R.C., Wang, M., Gao, Y., Tao, D.C., Li, X.L., Wu, X.D.: Image annotation by multiple-instance learning with discriminative feature mapping and selection. IEEE Trans. Cybern. **44**(5), 669–680 (2014)
8. Ilse, M., Tomczak, J., Welling, M.: Attention-based deep multiple instance learning. In: International Conference on Machine Learning, pp. 2127–2136 (2018)
9. Shi, X.S., Xing, F.Y., Xie, Y.P., Zhang, Z.Z., Cui, L., Yang, L.: Loss-based attention for deep multiple instance learning. In: Association for the Advancement of Artificial Intelligence, pp. 5742–5749 (2020)
10. Tarragó, D.S., Cornelis, C., Bello, R., Herrera, F.: A multi-instance learning wrapper based on the Rocchio classifier for web index recommendation. Knowl.-Based Syst. **59**, 173–181 (2014)
11. Wang, X.G., Yan, Y.L., Tang, P., Bai, X., Liu, W.Y.: Revisiting multiple instance neural networks. Pattern Recogn. **74**, 15–24 (2016)
12. Wei, X.S., Wu, J.X., Zhou, Z.H.: Scalable algorithms for multi-instance learning. IEEE Trans. Neural Netw. Learn. Syst. **28**(4), 975–987 (2017)
13. Wu, J., Pan, S., Zhu, X., Zhang, C., Wu, X.: Multi-instance learning with discriminative bag mapping. IEEE Trans. Knowl. Data Eng. **30**(6), 1065–1080 (2018)
14. Xu, B.C., Ting, K.M., Zhou, Z.H.: Isolation set-kernel and its application to multi-instance learning. In: Special Interest Group on Knowledge Discovery and Data Mining, pp. 941–949 (2019)
15. Yang, M., Zhang, Y.X., Wang, X.Z., Min, F.: Multi-instance ensemble learning with discriminative bags. IEEE Trans. Syst.Man Cybern. Syst., 1–12 (2021)

16. Zhang, M.L., Zhou, Z.H.: Multi-instance clustering with applications to multi-instance prediction. Appl. Intell. **31**(1), 47–68 (2009)
17. Zhang, W.J., Liu, L., Li, J.Y.: Robust multi-instance learning with stable instances, pp. 1682–1689. arXiv:1902.05066 (2020)
18. Zhou, Z.H., Jiang, K., Li, M.: Multi-instance learning based web mining. Appl. Intell. **22**, 135–147 (2005)
19. Zhou, Z.H., Sun, Y.Y., Li, Y.F.: Multi-instance learning by treating instances as non-I.I.D. samples. In: International Conference on Machine Learning, pp. 1249–1256 (2009)