Proceedings of Machine Learning Research 95:662-677, 2018 ACML 2018

Deep Multi-instance Learning with Dynamic Pooling

Yongluan Yan YONGLUANYAN@HUST.EDU.CN
Xinggang Wang XGWANG@HUST.EDU.CN
School of EIC, Huazhong University of Science and Technology

Xiaojie Guo XJ.MAX.GUO@GMAIL.COM
School of Computer Software, Tianjin University

Jiemin Fang JIEMING_FONG@HUST.EDU.CN
Wenyu Liu LIUWY@HUST.EDU.CN
School of EIC, Huazhong University of Science and Technology

Junzhou Huang JZHUANG75@GMAIL.COM
Tencent AI Lab & Department of CSE, University of Texas at Arlington

Editors: Jun Zhu and Ichiro Takeuchi

Abstract

End-to-end optimization of multi-instance learning (MIL) using neural networks is an important problem with many applications, in which a core issue is how to design a permutation-invariant pooling function without losing much instance-level information. Inspired by the dynamic routing in recent capsule networks, we propose a novel dynamic pooling function for MIL. It is an adaptive scheme for both key instance selection and modeling the contextual information among instances in a bag. The dynamic pooling iteratively updates the instance contribution to its bag. It is permutation-invariant and can interpret instance-to-bag relationship. The proposed dynamic pooling based multi-instance neural network has been validated on many MIL tasks and outperforms other MIL methods.

Keywords: Neural Network, Multi-instance Learning

1. Introduction

Weakly supervised learning (WSL), which aims to significantly reduce human annotation efforts, is an important problem in machine learning and has many applications in various domains. Referring to the definition in Zhou (2017), multi-instance learning (MIL) is a typical WSL, the training data of which are given with only coarse-grained labels. Originally, MIL was firstly introduced for the task of drug activity prediction (Dietterich et al. (1997)), and now it has been successfully applied to a wide spectrum of machine learning tasks, such as object detection (Tang et al. (2018, 2017); Wang et al. (2015); Cao et al. (2017)), semantic segmentation (Pathak et al. (2015, 2014)), scene classification (Wang et al. (2013)), text classification (Zafra et al. (2009); Zhou et al. (2009)), and medical diagnosis (Manivannan et al. (2017); Quellec et al. (2017)).

In MIL, each training sample is in a form of bag that contains a set of instances. Only bag-level labels are available. The goal of MIL is to train a classifier that predicts the label of a new bag. In this paper, we emphasize the case of the binary multiple instance classification (belong to the target class or not). The relationship between instances and

bags plays a major role in solving MIL problem. There existed different multiple instance (MI) assumptions that define relationships for various MIL applications. The standard MI assumption is that a bag is positive only if it contains at least one positive instance and otherwise is negative. Many methods such as EM-DD (Zhang and Goldman (2002)), mi-SVM (Andrews et al. (2003)), and VF (Liu et al. (2012)) are under this assumption and stress on "key instance selection" that triggers the bag label. However, the key-instance based MI assumption may be inappropriate in some domains. A simple example is given by Foulds and Frank (2010) as follows. For the beach or non-beach image classification problem, each image is described as a bag where instances are sand segment, sea segment, and other segments. If a bag belongs to the class of beach, instances of sand and sea must co-occur. If only an instance of sand or sea appears in the image, the class is still of non-beach. In this situation, the methods under the key-instance based MI assumption fails. Extending to more complex cases, generalized MI assumptions (Foulds and Frank (2010); Frank and Xu (2003); Li and Vasconcelos (2015) are raised to model the contextual information among instances for bag labels prediction.

For the sake of end-to-end optimization, neural networks (Ilse et al. (2018); Ramon and De Raedt (2000); Wang et al. (2018); Zhang and Zhou (2004); Zhou and Zhang (2002)) are effective to solve MIL problems. Ramon and De Raedt (2000) firstly applied neural networks to estimate instance probabilities and calculated bag probabilities by a log-sum-exp operator. Zhou and Zhang (2002) proposed a similar network called BP-MIP which replaces log-sum-exp operator with the max operator. Zhang and Zhou (2004) improved them with two extensions BP-MIP-DD and BP-MIP-PCA that are combined with Diverse Density (Maron and Lozano-Pérez (1998)) and PCA (Wold et al. (1987)) respectively. Unlike the mentioned works that focus on inferring instance labels, the MI-Net (Wang et al. (2018)) was proposed to pay more attention to learning the bag embedding. Following this pipeline, Ilse et al. (2018) incorporated an attention mechanism to learn the contribution of each instance to its bag embedding. As MI data are unordered, MIL methods with neural networks should be under the fundamental theorem of Permutation-invariant Symmetric Function (Qi et al. (2017); Zaheer et al. (2017)). In Ilse et al. (2018) and Wang et al. (2018), the whole process can be decomposed into three steps: (i) learning an instance embedding by the instance transformer, (ii) performing a permutation-invariant MIL pooling to generate a bag embedding, (iii) classifying a bag based on the bag embedding.

However, the previous permutation-invariant MIL pooling functions are hard to model the contextual information in a bag, because they are either predefined (such as max pooling) or ignoring the influence of other instances in the same bag (such as attention-based pooling). Inspired by the Capsule Networks (Sabour et al. (2017)), a dynamic pooling scheme is proposed in this paper. It learns the instance-to-bag relationship so as to generalize to various MI assumptions and keeps permutation invariance. Specifically, the dynamic pooling iteratively updates the instance contribution to its bag embedding during each feed forward time. Based on these instance contributions, the dynamic pooling highlights the key instance and models the contextual information among instances. The whole multi-instance neural network is optimized by the margin loss in an end-to-end manner. Therefore, we name it as the Dynamic Pooling for Multi-Instance Neural Network (DP-MINN).

In summary, we reveal that an adaptive scheme, jointly selecting the key instance and modeling the contextual information among instances, is helpful for MIL. Toward this end,

a novel dynamic pooling algorithm motivated by the dynamic routing in the capsule networks is designed. Besides the abilities to highlight the key instance and model contextual information, the dynamic pooling function inherits the merits of permutation invariance and makes instance-to-bag relationship interpretable. Thanks to the above advantages, our DP-MINN outperforms other MIL methods on many MIL tasks.

2. Related Work

2.1. Multiple Instance Learning

During the past decade, many MIL methods have been proposed, which can be roughly divided into three groups as follows Amores (2013): instance-space paradigm, bag-space paradigm, and embedded-space paradigm. Instance-space paradigm assumes the existence of hidden instance labels, so it builds a model to predict instance labels and aggregates them into bag labels under MI assumptions, such as mi-SVM (Andrews et al. (2003)), EM-DD (Zhang and Goldman (2002)), MIBoosting (Xu and Frank (2004)). Bag-space paradigm, like MInd (Cheplygina et al. (2015)), and mi-Graph (Zhou et al. (2009)), relies on the relationship between bags, treats bags as a whole, and then determines bag labels via nearest neighbor method or SVM to work directly in the original bag space. And embedded-space paradigm encodes a bag into the vocabulary-based feature and then converts the MIL problem to standard binary classification problem.

As instance labels are unavailable in MIL problems, MI assumptions are crucial to defining instance-to-bag relations. In Dietterich et al. (1997), the standard MI assumption was defined and aimed at picking the key instance which determines its bag label. Under the standard MI assumption, Zhang and Goldman (2002) improved the original DD algorithm using the EM approach and Liu et al. (2012) proposed a voting framework solution to predict bag labels. However, when extending to more complex cases, key instance detection may fail. So many works concentrate on modeling the contextual information among instances Zhou et al. (2009) studied inter-correlation of instances and raised two MIL methods, and Li and Vasconcelos (2015) considered a more general definition of MIL that both positive and negative bags are under soft constraints.

2.2. Capsule Network

Although convolutional neural networks (CNN) have shown remarkable performance on many computer vision tasks, it neglects important spatial hierarchies between simple and complex objects. Recently, Sabour et al. (2017) introduced a new architecture called Capsule Networks. Instead of neurons in CNN, all important information such as pose, deformation, and texture is stored in capsules which are described in a form of vectors. It uses the routing-by-agreement mechanism: the capsule vector is routed by parent capsules in the layer above and is iteratively updated by the agreement that is the scalar product of the capsule's prediction with each parent's output. The whole procedure is called the dynamic routing. Then Hinton et al. (2018) proposed a new type of capsule system that each capsule is represented by a 4×4 matrix and routing algorithm is replaced with EM algorithm. Besides these works, Jaiswal et al. (2018) raised CapsuleGAN as a framework

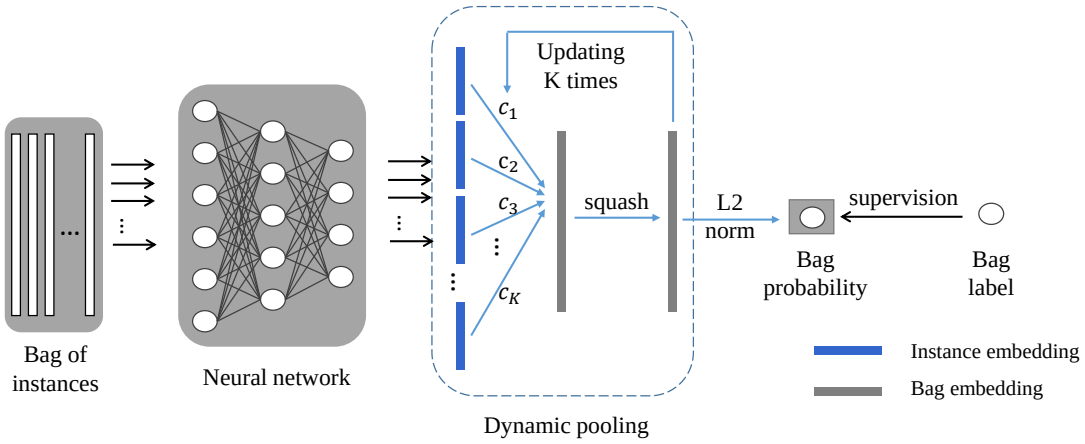

Figure 1: The architecture of Dynamic pooling for Multi-Instance Neural Network.

that incorporates capsules within GAN, and LaLonde and Bagci (2018) extended capsule networks to object segmentation.

Motivated by the routing-by-agreement idea, we proposed a dynamic pooling scheme via pooling-by-agreement which fully utilizes the instance-to-bag relationship. The differences between the dynamic routing and the dynamic pooling will be illustrated in Sec 3.5.

3. Dynamic pooling for multi-instance neural network

In this section, we firstly review the formulation of MIL, then introduce our DP-MINN, and lastly give a further discussion about it. Figure 1 gives the overall architecture of the DP-MINN.

3.1. Multiple instance learning

Problem formulation. MIL concentrates on handling the complex data in the form that each bag $X = \{x_1, x_2, ..., x_K\}$ is associated with multiple instances, where x_i is a d-dimensional feature vector and represents the i-th instance of the bag. The bag size K is various to different bags. Unlike the supervised learning, only bag label $Y \in \{0, 1\}$ is at hand, whereas the individual labels for instances are never reported. MIL aims to train a bag classifier to predict the label of a new bag. As introduced in Sec 1, instance-to-bag relationships are various under different MI assumptions. Hence, we do not represent a fixed MI assumption between instance labels and bag labels as previous MIL works. Instead, we stress to build a MIL model to predict bag labels.

Permutation invariance. Unlike pixels themselves having the spatial relation, in MIL, instances of a bag are a set of features without a specific order. Therefore, one important property of MI data is the invariance to input permutation. Under the fundamental theorem of symmetric functions (Zaheer et al. (2017); Qi et al. (2017)), any permutation-invariant symmetric functions M can be decomposed in the following form:

$$M(X) = \rho(\sum_{x \in X} \phi(x)). \tag{1}$$

where ρ and ϕ are suitable transformations.

MIL with neural networks. Both the MI-Net (Wang et al. (2018)) and the Attention Net (Ilse et al. (2018)) contain three steps: (i) learning an instance embedding by the instance transformer; (ii) performing a permutation-invariant MIL pooling to generate a bag embedding; (iii) classifying a bag based on the bag embedding. Each step has the permutation-invariant property following the fundamental theorem of symmetric functions (Zaheer et al. (2017); Qi et al. (2017)). The permutation-invariant MIL pooling is the key step because it bridges the instance space and the bag space. And it keeps the permutation invariance at the same time. The MI-Net (Wang et al. (2018)) proposes three predefined pooling functions (max pooling, mean pooling, and log-sum-exp pooling), and the Attention Net raises two flexible pooling functions (attention pooling and gated-attention pooling) based on attention mechanism.

3.2. Dynamic pooling

However, in the previous MIL pooling methods, it is hard to model the contextual information between the instances in a bag, since the pooling functions are feed-forward processes and the instance weights are computed individually. Motivated by the routing-by-agreement idea in the capsule networks, we propose a pooling-by-agreement scheme which is called dynamic pooling.

To illustrate clearly, we denote the instance transformer $f(\cdot)$ and instance embeddings $f(X) = \{f(x_1), f(x_2), ..., f(x_K)\}$ corresponding to the bag X. Our dynamic pooling can be expressed as a form of weighted-sum pooling as follows:

$$\sigma(X) = \sum_{i=1}^{K} c_i f(x_i), \tag{2}$$

where the instance weight c_i is a scalar which describes the contribution of the i-th instance to its bag embedding. Based on these weights, we follow Eq. (2) to aggregate instance embeddings into a bag embedding in a weighted-sum pooling fashion and use a non-linear "squashing" function above the bag embedding. The squashing function can be presented as Eq. (3), which ensures that short vectors get shrunk to almost zero length and long vectors get shrunk to a length slightly below 1.

$$s(X) = \frac{||\sigma(X)||^2}{1 + ||\sigma(X)||^2} \frac{\sigma(X)}{||\sigma(X)||}. \tag{3}$$

Different from previous MIL pooling functions, our instance weight c_i is calculated in a dynamic fashion. To describe the dynamic pooling process, we define a temporary instance weight denoted as b_i. Then, the instance weight c_i is determined by a simple *softmax* function as follows.

$$c_i = \frac{\exp(b_i)}{\sum_j \exp(b_j)}. \tag{4}$$

For clarify, the superscript t represents in the t-th iteration. Initially ($t = 1$), $b_i^1 = 0$ means each instance contributes equally to the bag embedding. Then, the instance weights are iteratively updated by considering their similarities to the latest bag embedding. The

Algorithm 1 Dynamic Pooling

Require: For each bag, instance embedding $\{f(x_i)\}$, and iteration times T.
Ensure: Bag embedding $s(X)$

```
for all instances: weight b_i^1 ← 0
for t = 1 to T do
    ∀i ∈ [1, K], c_i^t ← softmax(b_i^t)
    σ^t(X) ← Σ_i c_i^t f(x_i)
    s^t(X) ← squash(σ^t(X))
    ∀i ∈ [1, K], b_i^{t+1} ← b_i^t + f(x_i) · s^t(X)
end for
return s^T(X)
```

scalar product is used to calculate their similarities. Concretely, in the t-th iteration, we have the bag embedding $s^t(X)$; then the temporary instance weight b_i^t is updated as follows:

$$b_i^{t+1} = b_i^t + f(x_i) \cdot s^t(X). \tag{5}$$

Different from the previous feed-forward pooling functions, in the dynamic pooling procedure, the bag embedding which is an integration of multiple instances can backward induce the instance weights. In addition, the *softmax* function Eq. (4) enforces instances compete to each other. In this way, the contextual information between bags can be modeled. The overall dynamic pooling procedure for T times is illustrated in Algorithm 1.

3.3. Optimization

After each feed forward, we can obtain the bag embedding $s^T(X)$. And we do ℓ^2 norm over $s^T(X)$ to represent the probability of positive bag and denote it as $||s||$. In order to optimize our DP-MINN, we use the margin loss during training phase:

$$L(X) = Y \max(0, m^+ - ||s||)^2 + (1 - Y) \max(0, ||s|| - m^-)^2, \tag{6}$$

where $m^+ = 0.9$, $m^- = 0.1$, and Y is determined by the bag label. Besides, this network is optimized by the Adam (Kingma and Ba (2014)).

3.4. Permutation-invariant property of dynamic pooling

As mentioned in the Sec 3.1, any permutation-invariant symmetric function $M(X)$ can be composed as $\rho(\sum_{x \in X} \phi(x))$. In order to prove the permutation-invariant property of our dynamic pooling, we will illustrate that our pooling process fulfills the requirements of permutation-invariant symmetric function. The essence of our dynamic pooling is to do weighted-sum pooling. Different from the Attention Net (Ilse et al. (2018)) that weight is learned based on the instance itself, our weight considers other instances belonging to the same bag and its final value is determined by T times iterations.

At the initial time ($t = 1$), dynamic pooling begins with mean pooling:

$$\sigma^1(X) = \sum_i c_i^1 f(x_i), \text{ where } \forall i \in [1, K] \quad c_i^1 = \frac{1}{K}. \tag{7}$$

Clearly, mean pooling is a typical symmetric function. In the t-th iteration ($t > 1$), pooling function is as

$$\sigma^t(X) = \sum_i c_i^t \cdot f(x_i) = \sum_i \text{softmax}(\sum_{t>1} f(x_i) \cdot s^{t-1}(X)) f(x_i). \tag{8}$$

And $s^t(X)$ is the bag embedding at the t-th iteration, which is an output of the symmetric function and keeps the permutation-invariant property. Based on the decomposed form the symmetric function in Eq. (1), we can regard the dynamic process is part of ϕ. Then ℓ^2 norm which represents the position of ρ computes the length of bag X as the bag probability.

3.5. Difference with the dynamic routing in Capsule network

Both the dynamic pooling and the dynamic routing consider the part-to-whole relationship. They learn weights processed by the *softmax* function and then perform the weighted-sum pooling. However, it is worth to note that the meaning of weights is different. That is to say, the role of the *softmax* function is not the same. In the dynamic routing, the *softmax* function is applied to weights of all parent capsules to one of the child capsules. So each weight means that the ratio of the corresponding capsule in the layer above is sent to the child capsule. But in the dynamic pooling, the weight describes the instance contribution to the bag embedding. The *softmax* function is performed over all instance contributions of the same bag and lets them interact.

4. Experiment

In this section, we evaluate our DP-MINN on various MIL tasks, including drug activity prediction, localized content-based image retrieval, text categorization, and medical diagnosis.

4.1. Datasets

MUSK1 and MUSK2 (Dietterich et al. (1997)) are typical MIL datasets for drug activation prediction. We regard molecules and their different conformations as bags and instances, respectively. Each conformation is described as a 166-dimensional feature vector. In MUSK1, there are 47 positive bags and 45 negative bags; MUSK2 is much bigger including 49 positive bags and 63 negative bags.

Fox, Tiger, and Elephant (Andrews et al. (2003)) are other widely used MIL benchmarks to identify whether a given image contains target animal or not. Bags are images and instances are corresponding image segments. Positive bags are composed of the target animal class, and negative bags are randomly chosen from other animal classes. Moreover, each instance is described as a 230-dimensional feature containing color, texture, and shape information of associative image segment.

20 Newsgroups (Zhou et al. (2009)) is a text-categorization dataset of 20 different news groups. As news articles contain multiple paragraphs of different topics, we can naturally regard the text categorization problem as a MIL problem. Bags are articles and instances are paragraphs which are preprocessed by TF-IDF. Each category is composed

Table 1: Comparison results (mean±*standard deviations of mean*) of different methods for bag classification on MUSK1, MUSK2, Fox, Tiger, and Elephant datasets (task: drug activation prediction and localized content-based image retrieval).

Dataset	MUSK1	MUSK2	Fox	Tiger	Elephant
mi-SVM	$0.874 \pm N/A$	$0.836 \pm N/A$	$0.582 \pm N/A$	$0.784 \pm N/A$	$0.822 \pm N/A$
MI-SVM	$0.779 \pm N/A$	$0.843 \pm N/A$	$0.578 \pm N/A$	$0.840 \pm N/A$	$0.843 \pm N/A$
MI-Kernel	0.880 ± 0.031	0.893 ± 0.015	0.603 ± 0.028	0.842 ± 0.010	0.843 ± 0.016
EM-DD	0.849 ± 0.044	0.869 ± 0.048	0.609 ± 0.045	0.730 ± 0.043	0.771 ± 0.043
mi-Graph	0.889 ± 0.033	*0.903 ± 0.039*	0.620 ± 0.044	*0.860 ± 0.037*	*0.869 ± 0.035*
miVLAD	0.871 ± 0.043	0.872 ± 0.042	0.620 ± 0.044	0.811 ± 0.039	0.850 ± 0.036
miFV	**0.909 ± 0.040**	0.884 ± 0.042	0.621 ± 0.049	0.813 ± 0.037	0.852 ± 0.036
MI-Net	0.887 ± 0.041	0.859 ± 0.046	*0.622 ± 0.038*	0.830 ± 0.032	0.862 ± 0.034
Att. Net	0.892 ± 0.040	0.858 ± 0.048	0.615 ± 0.043	0.839 ± 0.022	0.868 ± 0.022
Gated Att. Net	0.900 ± 0.050	0.863 ± 0.042	0.603 ± 0.029	0.845 ± 0.018	0.857 ± 0.027
DP-MINN	*0.907 ± 0.036*	**0.926 ± 0.043**	**0.655 ± 0.052**	**0.897 ± 0.028**	**0.894 ± 0.030**

of 50 positive bags and 50 negative bags. And each positive bag contains 3% posts from the target news groups, whereas negative bags choose their instances randomly from other news groups.

UCSB breast (Kandemir et al. (2014)) and Messidor (Decencière et al. (2014); Kandemir and Hamprecht (2014)) are two widely used datasets in computer-aided medical diagnosis. UCSB breast dataset is taken from 32 benign (negative) and 26 malignant (positive) breast cancer patients. Bags are the whole cancer images, and instance are 708-dimensional features processed by histogram, LBP, SIFT descriptor in 7×7 patches. Messidor contains $1,200$ eye fundus images from 654 diabetes (positive) and 546 healthy patients. Similarly, bags are whole image and instances are 100-dimensional features reduced by PCA (Wold et al. (1987)).

More detailed information about these MIL datasets is summarized in the supplementary material.

4.2. Experimental Setup

In our experiments, we follow the MI-Net (Wang et al. (2018)) and use the same architecture. Our network is composed of three fully connected layers with 256, 128, 64 neurons and a dynamic pooling function. Weights of fully connected layers are initialized by a truncated normal distribution and biases are initialized to be 0. And iteration times T of the dynamic pooling is assigned to 3. We adopt the Adam (Kingma and Ba (2014)) to optimize our network. Detailed hyper-parameters of the optimization process such as learning rate, weight decay and decay scheme are listed in the supplementary material. We run 5 times 10-fold cross validation independently and report average results as final results. Our code is written in Python, based on TensorFlow (Abadi et al. (2015)). Experiments are run on a PC with Inter(R) i7-4790K CPU (4.00GHZ) and 32GB RAM.

Table 2: Comparison results (mean±*standard deviations of mean*) of different methods for bag classification on 20 Newsgroups (task: text categorization).

Dataset	MI-Kernel	mi-Graph	miFV	MI-Net	Att. Net	Gated Att. Net	DP-MINN
alt.atheism	0.602 ± 0.039	0.655 ± 0.040	*0.848 ± 0.053*	0.776 ± 0.045	0.784 ± 0.084	0.780 ± 0.074	**0.896 ± 0.041**
comp.graphics	0.470 ± 0.033	0.778 ± 0.016	0.594 ± 0.063	*0.826 ± 0.060*	0.774 ± 0.081	0.764 ± 0.073	**0.858 ± 0.048**
comp.win.misc	0.510 ± 0.052	0.631 ± 0.015	0.615 ± 0.077	0.678 ± 0.045	0.686 ± 0.088	*0.700 ± 0.080*	**0.794 ± 0.051**
comp.ibm.pc.hw	0.469 ± 0.036	0.696 ± 0.027	0.665 ± 0.066	**0.778 ± 0.058**	0.632 ± 0.087	0.640 ± 0.080	*0.770 ± 0.071*
comp.sys.mac.hw	0.445 ± 0.032	0.617 ± 0.048	0.660 ± 0.070	*0.792 ± 0.051*	0.744 ± 0.084	0.754 ± 0.082	**0.860 ± 0.058**
comp.win.x	0.508 ± 0.043	0.698 ± 0.021	0.768 ± 0.069	*0.786 ± 0.050*	0.766 ± 0.093	0.780 ± 0.075	**0.878 ± 0.051**
misc.forsale	0.518 ± 0.025	0.698 ± 0.021	0.565 ± 0.065	0.652 ± 0.057	*0.706 ± 0.076*	0.674 ± 0.072	**0.787 ± 0.061**
rec.autos	0.529 ± 0.033	0.720 ± 0.037	0.667 ± 0.074	*0.774 ± 0.054*	0.762 ± 0.081	0.724 ± 0.091	**0.838 ± 0.045**
rec.motorcycle	0.506 ± 0.035	0.640 ± 0.028	0.802 ± 0.064	0.762 ± 0.052	0.750 ± 0.097	*0.814 ± 0.066*	**0.890 ± 0.050**
rec.sport.baseball	0.517 ± 0.028	0.647 ± 0.031	0.779 ± 0.066	**0.856 ± 0.051**	0.774 ± 0.080	0.790 ± 0.078	**0.856 ± 0.047**
rec.sport.hockey	0.513 ± 0.034	0.850 ± 0.025	0.823 ± 0.061	0.862 ± 0.038	**0.936 ± 0.041**	*0.932 ± 0.045*	0.929 ± 0.040
sci.crypt	0.563 ± 0.036	0.696 ± 0.039	0.760 ± 0.065	0.694 ± 0.064	*0.804 ± 0.063*	0.748 ± 0.088	**0.854 ± 0.047**
sci.electron	0.506 ± 0.020	0.871 ± 0.017	0.555 ± 0.070	*0.930 ± 0.040*	0.854 ± 0.053	0.828 ± 0.064	**0.932 ± 0.036**
sci.med	0.506 ± 0.019	0.621 ± 0.039	0.783 ± 0.056	*0.818 ± 0.047*	0.772 ± 0.090	0.742 ± 0.101	**0.846 ± 0.052**
sci.space	0.547 ± 0.025	0.757 ± 0.034	0.818 ± 0.059	0.752 ± 0.050	0.888 ± 0.062	*0.894 ± 0.063*	**0.908 ± 0.047**
soc.religion.chri	0.492 ± 0.034	0.590 ± 0.047	*0.814 ± 0.062*	0.782 ± 0.051	0.726 ± 0.088	0.708 ± 0.100	**0.840 ± 0.052**
talk.polit.guns	0.477 ± 0.038	0.585 ± 0.060	*0.747 ± 0.067*	0.652 ± 0.052	0.714 ± 0.074	0.708 ± 0.078	**0.822 ± 0.049**
talk.polit.mideast	0.559 ± 0.028	0.736 ± 0.026	0.793 ± 0.060	*0.794 ± 0.057*	0.750 ± 0.084	0.784 ± 0.064	**0.830 ± 0.047**
talk.polit.misc	0.515 ± 0.037	0.704 ± 0.036	0.697 ± 0.068	0.654 ± 0.060	0.788 ± 0.091	*0.806 ± 0.078*	**0.822 ± 0.051**
talk.religion.misc	0.554 ± 0.043	0.633 ± 0.035	0.739 ± 0.068	0.700 ± 0.051	0.738 ± 0.074	*0.746 ± 0.082*	**0.814 ± 0.045**
average	0.515	0.679	0.726	*0.820*	0.767	0.766	**0.851**

4.3. Experimental results

Drug Activation Prediction. Table 1 presents the results (mean accuracies and the standard deviation of mean) of our method and other competing MIL methods. Due to the lack of standard deviation from mi-SVM (Andrews et al. (2003)), and MI-Kernel (Zhou et al. (2009)), we use N/A to represent the absence. We highlight the top two best results in bold and italic respectively. From the Table 1, miFV (Wei et al. (2017)) goes to the best accuracy 90.9% and Gated-attention Net also reaches around 90.9% on MUSK1. Our DP-MINN achieves the second best result 90.7% which is slightly lower than miFV. But it outperforms Attention Net (Ilse et al. (2018)) and MI-Net (Wang et al. (2018)) by around 1.7% and 2.3%. On MUSK2, the efficacy of our DP-MIINN is proved further. Our method gets the best accuracy 92.6%. Compared to other MIL methods with neural networks (MI-Net, Attention Net, and Gated-attention Net), our DP-MINN acts much more stable. Therefore, it indicates the performance of our dynamic pooling.

Localized Content-based Image Retrieval. In the localized content-based image retrieval task, we conduct experiments on three animal MIL datasets (Fox, Tiger, and Elephant). Results are in the last three columns of the table 1 and shows the state-of-the-art results on these MIL datasets. Mean accuracies has been improved at least 3.7%, in contrast to other MIL methods with neural networks.

Text Categorization. In the task of text classification on the 20 Newsgroups dataset, our DP-MINN also outperforms the other methods by a large margin. Table 2 lists the average accuracies, from which we can find that our method wins over the competitors for all the cases except for the *comp.ibm.pc.hardware* and *rec.sport.hockey*. Although for *comp.ibm.pc.hardware* and *rec.sport.hockey* datasets, our DP-MINN achieves the competing result 77.0% and 92.9%, respectively. And the average accuracies of all 20 MIL datasets

Table 3: Comparison results (mean±*standard deviations of mean*) of different methods for bag classification on UCSB breast and Messidor dataset (task: medical diagnosis).

Methods	MI-SVM	miFV	MI-Net	Att. Net	Gated Att. Net	DP-MINN
UCSB breast	*0.911 ± 0.016*	0.870 ± 0.050	0.806 ± 0.104	0.883 ± 0.062	0.887 ± 0.066	**0.927 ± 0.070**
Messidor	0.640 ± 0.050	0.715 ± 0.047	*0.731 ± 0.018*	0.703 ± 0.041	0.698 ± 0.048	**0.740 ± 0.020**

indicate that our method and MI-Net outperforms others, including MI-Kernel, mi-Graph (Zhou et al. (2009)), miFV (Wei et al. (2017)), Attention Net, Gated-attention net (Ilse et al. (2018)), with about 10% improvement in performance. Besides, the average accuracy of our DP-MINN is better than MI-Net by 3.7%, respectively.

Medical Diagnosis. Here we present our results for medical diagnosis which is another hot MIL application recently. The UCSB breast dataset and Messidor dataset are two public computer-aided medical diagnosis dataset. In Table 3, it shows that our DP-MINN obtains the best results on both the UCSB breast dataset and the Messidor dataset compared to other MIL methods. Besides, the results show that the original MI-Net does not work well in this task. Especially, on the UCSB breast dataset, MI-Net works much worse than the traditional miFV and MI-SVM methods. The attention mechanism helps to improve the results of MI-Net. However, the improvement of attention network is not consistent; the same phenomenon can also be observed from Table 2. DP-MINN consistently improve the results of MI-Net which confirms the effectiveness of the proposed dynamic pooling.

5. Ablation Study and Discussion

In this section, we carry out some ablation studies about the influence of the iteration times T and loss functions to our DP-MINN over MUSK1, MUSK2, Fox, Tiger, and Elephant datasets.

Different iteration times T. As mentioned in Sec 3, the dynamic pooling learns the instance contribution to its bag embedding by updating it for T times during each feed forward. When $T = 1$, the dynamic pooling degenerates to the mean pooling. With T increasing, instance contribution is changed considering the contextual information among instances in a bag. So we perform comparison experiments to study the influence of different iteration times. In Figure 2, we highlight results over different MIL datasets with various markers. And we also present the average results of all these datasets to reveal that performance of $T = 3$ is slightly better than other iteration times. Thus we recommend updating three times in the dynamic pooling.

Different loss functions. In previous MIL methods with neural networks, the cross entropy loss is one of the most popular loss functions. Instead, Our DP-MINN uses the margin loss during training. To discuss about the impact of different loss functions, we perform detailed comparison between the margin loss and the cross entropy loss. To build our DP-MINN with the cross entropy loss, we change the architecture by applying the bag embedding which is before the squash function to a bag label predictor and then outputting the bag probability, following the MI-Net (Wang et al. (2018)). The bag label predictor is a new fully connected layer with only one neuron and sigmoid activation. Finally, combining

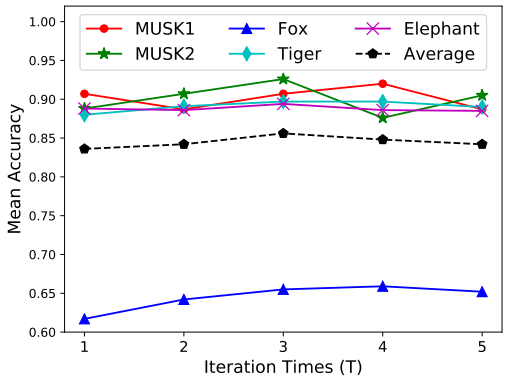

Figure 2: Comparisons of different iteration times T on five MIL benchmarks.

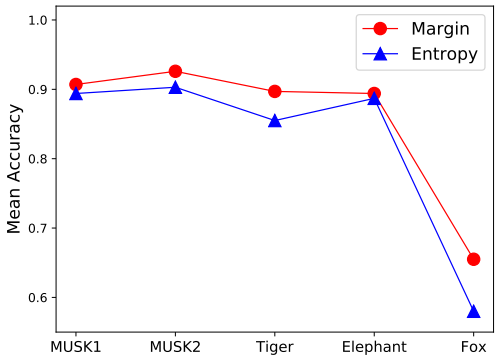

Figure 3: Comparisons of margin loss and cross entropy loss on five MIL benchmarks.

the bag label with the bag probability, we can calculate the cross entropy loss. Figure 3 indicates that on Tiger and Fox datasets the results based on the margin loss are better with a large margin and on other three datasets results still have slight improvement. Therefore, it proves the effectiveness of the margin loss to our DP-MINN.

6. Conclusion

In this paper, we propose a novel pooling function for multiple instance neural networks. The pooling function jointly selects the key instance and models the contextual information in a bag. Different from the dynamic routing method, our dynamic pooling method learns the instance-to-bag relation rather than capsule-to-capsule information. In addition, the dynamic pooling function is also permutation-invariant to the unordered instances in a bag. In the experiments, various MIL tasks, including image classification, text classification, and medical diagnose, have been investigated. The results show that the proposed DP-MINN outperforms other MIL methods, including the recent attentive pooling methods for multiple instance neural networks.

As stressed in capsule networks, the pooling function is an important part in neural network. However, traditional max pooling may lose lots of useful information of input examples. The routing mechanisms in capsule networks are designed to preserve more information. In this work, though we focus on the multiple instance neural network, the results show that a dynamic pooling-by-agreement method can significantly improve the performance. The result confirms the necessity of developing more comprehensive pooling function for neural networks. In the future, we would like to explore dynamic pooling function more neural network in different applications.

Acknowledgments

This work was partly supported by NSFC (No. 61733007, No. 61876212 No. 61503145, & No. 61572207). Xinggang Wang was sponsored by CCF-Tencent Open Research Fund, Hubei Scientific and Technical Innovation Key Project, and the Program for HUST Academic Frontier Youth Team.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL https://www.tensorflow.org/. Software available from tensorflow.org.

Jaume Amores. Multiple instance classification: Review, taxonomy and comparative study. *Artificial Intelligence*, 201:81–105, 2013.

Stuart Andrews, Ioannis Tsochantaridis, and Thomas Hofmann. Support vector machines for multiple-instance learning. In *Advances in neural information processing systems*, pages 577–584, 2003.

Liujuan Cao, Feng Luo, Li Chen, Yihan Sheng, Haibin Wang, Cheng Wang, and Rongrong Ji. Weakly supervised vehicle detection in satellite images via multi-instance discriminative learning. *Pattern Recognition*, 64:417–424, 2017.

Veronika Cheplygina, David MJ Tax, and Marco Loog. Multiple instance learning with bag dissimilarities. *Pattern Recognition*, 48(1):264–275, 2015.

Etienne Decencière, Xiwei Zhang, Guy Cazuguel, Bruno Lay, Béatrice Cochener, Caroline Trone, Philippe Gain, Richard Ordonez, Pascale Massin, Ali Erginay, Béatrice Charton, and Jean-Claude Klein. Feedback on a publicly distributed image database: the Messidor database. *Image Analysis and Stereology*, pages 231–234, 2014.

Thomas G Dietterich, Richard H Lathrop, and Tomás Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial intelligence*, 89(1-2):31–71, 1997.

James Foulds and Eibe Frank. A review of multi-instance learning assumptions. *The Knowledge Engineering Review*, 25(1):1–25, 2010.

Eibe Frank and Xin Xu. Applying propositional learning algorithms to multi-instance data. 2003.

Geoffrey Hinton, Nicholas Frosst, and Sara Sabour. Matrix capsules with em routing. 2018.

Maximilian Ilse, Jakub M Tomczak, and Max Welling. Attention-based deep multiple instance learning. *arXiv preprint arXiv:1802.04712*, 2018.

Ayush Jaiswal, Wael AbdAlmageed, and Premkumar Natarajan. Capsulegan: Generative adversarial capsule network. *arXiv preprint arXiv:1802.06167*, 2018.

Melih Kandemir and Fred A Hamprecht. Computer-aided diagnosis from weak supervision: A benchmarking study. *Computerized Medical Imaging and Graphics, in press*, 2014. doi: 10.1016/j.compmedimag.2014.11.010.

Melih Kandemir, Chong Zhang, and Fred A Hamprecht. Empowering multiple instance histopathology cancer diagnosis by cell graphs. In *MICCAI*, 2014.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Rodney LaLonde and Ulas Bagci. Capsules for object segmentation. *arXiv preprint arXiv:1804.04241*, 2018.

Weixin Li and Nuno Vasconcelos. Multiple instance learning for soft bags via top instances. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4277–4285, 2015.

Guoqing Liu, Jianxin Wu, and Z-H Zhou. Key instance detection in multi-instance learning. In *Asian Conference on Machine Learning*, pages 253–268, 2012.

Siyamalan Manivannan, Caroline Cobb, Stephen Burgess, and Emanuele Trucco. Subcategory classifiers for multiple-instance learning and its application to retinal nerve fiber layer visibility classification. *IEEE transactions on medical imaging*, 36(5):1140–1150, 2017.

Oded Maron and Tomás Lozano-Pérez. A framework for multiple-instance learning. In *Advances in neural information processing systems*, pages 570–576, 1998.

Deepak Pathak, Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional multi-class multiple instance learning. *arXiv preprint arXiv:1412.7144*, 2014.

Deepak Pathak, Philipp Krahenbuhl, and Trevor Darrell. Constrained convolutional neural networks for weakly supervised segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 1796–1804, 2015.

Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 1(2):4, 2017.

Gwenolé Quellec, Guy Cazuguel, Béatrice Cochener, and Mathieu Lamard. Multiple-instance learning for medical image and video analysis. *IEEE reviews in biomedical engineering*, 10:213–234, 2017.

Jan Ramon and Luc De Raedt. Multi instance neural networks. In *Proceedings of the ICML-2000 workshop on attribute-value and relational learning*, pages 53–60, 2000.

Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In *Advances in Neural Information Processing Systems*, pages 3859–3869, 2017.

Peng Tang, Xinggang Wang, Xiang Bai, and Wenyu Liu. Multiple instance detection network with online instance classifier refinement. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2843–2851, 2017.

Peng Tang, Xinggang Wang, Angtian Wang, Yongluan Yan, Wenyu Liu, Junzhou Huang, and Alan Yuille. Weakly supervised region proposal network and object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 352–368, 2018.

Xinggang Wang, Baoyuan Wang, Xiang Bai, Wenyu Liu, and Zhuowen Tu. Max-margin multiple-instance dictionary learning. In *International Conference on Machine Learning*, pages 846–854, 2013.

Xinggang Wang, Zhuotun Zhu, Cong Yao, and Xiang Bai. Relaxed multiple-instance svm with application to object discovery. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1224–1232, 2015.

Xinggang Wang, Yongluan Yan, Peng Tang, Xiang Bai, and Wenyu Liu. Revisiting multiple instance neural networks. *Pattern Recognition*, 74:15–24, 2018.

Xiu-Shen Wei, Jianxin Wu, and Zhi-Hua Zhou. Scalable algorithms for multi-instance learning. *IEEE transactions on neural networks and learning systems*, 28(4):975–987, 2017.

Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

Xin Xu and Eibe Frank. Logistic regression and boosting for labeled bags of instances. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 272–281. Springer, 2004.

Amelia Zafra, Cristóbal Romero, Sebastián Ventura, and Enrique Herrera-Viedma. Multi-instance genetic programming for web index recommendation. *Expert Systems with Applications*, 36(9):11470–11479, 2009.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Ruslan R Salakhutdinov, and Alexander J Smola. Deep sets. In *Advances in Neural Information Processing Systems*, pages 3394–3404, 2017.

Min-Ling Zhang and Zhi-Hua Zhou. Improve multi-instance neural networks through feature selection. *Neural Processing Letters*, 19(1):1–10, 2004.

Qi Zhang and Sally A Goldman. Em-dd: An improved multiple-instance learning technique. In *Advances in neural information processing systems*, pages 1073–1080, 2002.

Zhi-Hua Zhou. A brief introduction to weakly supervised learning. *National Science Review*, 2017.

Zhi-Hua Zhou and Min-Ling Zhang. Neural networks for multi-instance learning. In *Proceedings of the International Conference on Intelligent Information Technology, Beijing, China*, pages 455–459, 2002.

Zhi-Hua Zhou, Yu-Yin Sun, and Yu-Feng Li. Multi-instance learning by treating instances as non-iid samples. In *Proceedings of the 26th annual international conference on machine learning*, pages 1249–1256. ACM, 2009.

Appendix A. Additional Experiments

This is the supplementary material to the paper "Deep Multi-instance Learning with Dynamic Pooling". Here we provide a detailed MIL dataset description, hyper-parameters of the optimization, and results of ablation study.

Dataset description In Table 4, we give a general description of all MIL datasets used in the experiments.

Table 4: Detailed Characteristics of the MIL datasets. "# positive" ("#negative") presents the number of positive(negative) bags used in each round. For Text category dataset, because it contains 20 sub-datasets, we present one of them in it.

Dataset	# attribute	# bag			# instance
		positive	negative	total	
MUSK1	166	47	45	92	476
MUSK2	166	39	63	102	6598
Elephant	230	100	100	200	1391
Fox	230	100	100	200	1320
Tiger	230	100	100	200	1220
alt.atheism	200	50	50	100	5443
UCSB breast	708	26	32	58	2002
Messidor	687	654	546	1200	12352

Additional details of optimization The hyper-parameters of the optimization procedure including learning rate, weight decay, max iterations, and decay step, are listed in Table 5. We adopt Adam with the exponential decay scheme in our training progresses. Concretely, for 20 Newsgroups datasets, learning rate is decayed every 64 iterations with a base of 0.96 and terminates at $5,000$ iterations; for other MIL datasets, learning rate is decayed every 250 iterations with the same base and terminates at $10,000$ iterations. The hyper-parameters we provide are determined by the model selection procedure for which the highest validation performance was achieved.

Additional results of ablation study We represent the results of ablation study over MUSK1, MUSK2, Fox, Tiger, and Elephant datasets. Table 6, we show the comparison

Table 5: The hyper-parameters of the optimization procedure

Dataset	Learning rate	Weight decay	Iterations	decay steps
MUSK1	0.0005	0.005	10,000	250
MUSK2	0.0005	0.005	10,000	250
Fox	0.001	0.01	10,000	250
Tiger	0.001	0.005	10,000	250
Elephant	0.001	0.005	10,000	250
20 Newsgroups	0.001	0.001	5,000	64
UCSB breast	0.0001	0.0001	10,000	250
Messidor	0.0005	0.001	10,000	250

results when the iteration times $T = 1, 2, ..., 5$. Besides, we highlight the top two best results in bold and italic, respectively. And results of our network with the margin loss and the cross entropy loss are listed in the Table 7.

Table 6: Comparison results (mean±*standard deviations of mean*) of different iteration times T for bag classification on MUSK1, MUSK2, Fox, Tiger, and Elephant datasets.

Iteration times	MUSK1	MUSK2	Fox	Tiger	Elephant
T=1	*0.907 ± 0.033*	0.888 ± 0.048	0.617 ± 0.047	0.880 ± 0.030	0.880 ± 0.032
T=2	0.887 ± 0.042	*0.907 ± 0.045*	0.642 ± 0.042	0.891 ± 0.029	*0.886 ± 0.036*
T=3	*0.907 ± 0.036*	**0.926 ± 0.043**	*0.655 ± 0.052*	*0.897 ± 0.028*	**0.894 ± 0.030**
T=4	**0.920 ± 0.030**	0.876 ± 0.058	**0.659 ± 0.047**	**0.899 ± 0.030**	*0.886 ± 0.035*
T=5	0.887 ± 0.035	0.905 ± 0.040	0.652 ± 0.042	0.890 ± 0.034	0.885 ± 0.032

Table 7: Comparison results (mean±*standard deviations of mean*) of different loss function for bag classification on MUSK1, MUSK2, Fox, Tiger, and Elephant datasets.

Loss function	MUSK1	MUSK2	Fox	Tiger	Elephant
Margin	**0.907 ± 0.040**	**0.926 ± 0.043**	**0.655 ± 0.052**	**0.897 ± 0.028**	**0.894 ± 0.030**
Cross entropy	0.894 ± 0.060	0.903 ± 0.044	0.580 ± 0.058	0.855 ± 0.042	0.887 ± 0.048