SCIENCE CHINA
Information Sciences

March 2021, Vol. 64 130102:1–130102:15
https://doi.org/10.1007/s11432-020-3117-3

• RESEARCH PAPER •
Special Focus on Weakly Supervised Learning

Deep multiple instance selection

Xin-Chun LI, De-Chuan ZHAN*, Jia-Qi YANG & Yi SHI

State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023*, China*

Received 26 June 2020/Accepted 30 July 2020/Published online 7 February 2021

Abstract Multiple instance learning (MIL) assigns a single class label to a bag of instances tailored for some real-world applications such as drug activity prediction. Classical MIL methods focus on figuring out interested instances, that is, region of interests (ROIs). However, owing to the non-differentiable selection process, these methods are not feasible in deep learning. Thus, we focus on fusing ROIs identification with deep MILs in this paper. We propose a novel deep MIL framework based on hard selection, that is, deep multiple instance selection (DMIS), which can automatically figure ROIs out in an end-to-end approach. To be specific, we propose DMIS-GS for instance selection via gumbel softmax or gumbel top-k, and then make predictions for this bag without the interference of redundant instances. For balancing exploration and exploitation of key instances, we apply a cooling down approach to the temperature in DMIS-GS, and propose a variance normalization method to make this hyper-parameter tuning process much easier. Generally, we give a theoretical analysis of our framework. The empirical investigations reveal the proposed frameworks' superiorities against classical MIL methods on generalization ability, positioning ROIs, and comprehensibility on both synthetic and real-world datasets.

Keywords multiple instance learning, instance selection, gumbel softmax, variance normalization, hard attention

Citation Li X-C, Zhan D-C, Yang J-Q, et al. Deep multiple instance selection. Sci China Inf Sci, 2021, 64(3): 130102, https://doi.org/10.1007/s11432-020-3117-3

1 Introduction

In common machine learning assumptions, instances and labels can be provided in a pairwise mechanism as full supervision signals. However, a class label is only provided for a bag of instances in some special applications, and multiple instance learning (MIL) [1, 2] is tailored for these problems. In MIL, a bag can be a molecule with different alternative low-energy shapes [1], a document containing many text segments [3], or an image with small regions [4]. The core difficulty of MIL lies in the missing labels for instances [5], and many MIL methods focus on mining the region of interests (ROIs).

Some classical methods are not feasible in deep networks, although they can locate ROIs, such as MI-SVM [6] and KI-SVM [7], mainly owing to the complex optimization strategy and the non-differentiable selection process. Some instance-level MILs [6, 8, 9] make predictions for each individual instance, which can be used to mine ROIs according to the predicted instance labels. Similarly, these approaches are not so easy to be applied to deep models. Also these approaches' performances are usually worse than bag-level ones [4] as shown in [5].

In deep MILs, aggregation functions are utilized to aggregate instance predictions or instance embeddings. While the max-pooling operator [4,10] is mostly used in MIL, the non-differentiable property limits the model's capacity. Thus, the attention based MIL [11] is utilized to weight instances in an end-to-end approach. Although the learned weights can figure ROIs out and make MIL easier to understand, it is originally designed for weighting instances and belongs to a kind of post-selection method. Compared to this soft attention aggregation process, we propose that the hard selection code can filter disturbing instances out, for example, the negative instances in the positive bag, which fits the essence of MIL perfectly. Some other similar mehthods utilize a clustering-based instance selection mechanism for MIL

* Corresponding author (email: zhandc@nju.edu.cn)

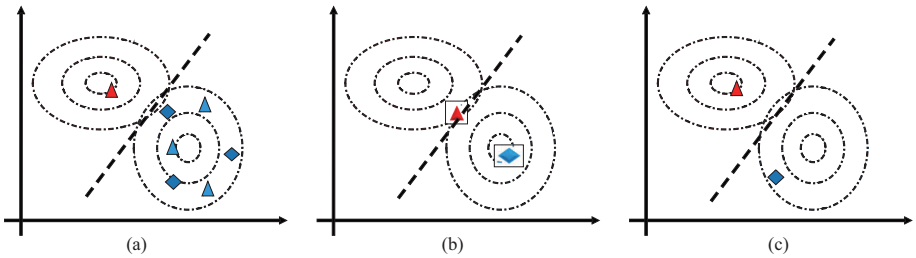

Figure 1 (Color online) Illustrations for MIL problem and the difference between soft aggregation and hard selection. The ellipses are the true distributions of positive (red) and negative (blue) instances, whereas the black dotted line is the oracle classifier. Triangles are instances from a positive bag, whereas diamonds are from a negative bag. (a) Standard assumption in MIL. (b) Soft aggregation in MIL, the weighted positive and negative bags are plotted, and the positive bag will be wrongly classified with the interference of negative instances. (c) Hard selection in MIL, the selected instances in each bag are plotted, and the interference of negative instances in the positive bag can be omitted.

as in [12], which can also select key instances out. However, we focus on the attention-based deep MIL methods in this paper.

We propose deep multiple instance selection (DMIS) in this study, from which interested instances can be identified and selected in the learning process. In order to optimize the hard selection process, we utilize gumbel softmax or gumbel top-k for end-to-end training, and the method is named as DMIS-GS. For exploring ROIs as much as possible, we initially apply a higher temperature to DMIS-GS, and then, we gradually decay it for better exploitation. We also propose a variance normalization approach to make this temperature tuning process much easier and more transferable across tasks. Hence, our work makes four contributions:

• We enable deep MILs to identify ROIs in the learning process via an end-to-end approach, and this is the first attempt to explicitly boost deep MILs with hard instance selection process as far as we know.

• Utilizing gumbel softmax or gumbel top-k for optimizing the hard selection process, we propose a novel deep MIL framework DMIS, and a simple variance normalization trick is also proposed in the temperature annealing process.

• We give a theoretical analysis of our framework and show that it is universal to be applied to other set problems.

• Abundant empirical studies are investigated, which verify that the proposed DMIS-GS can obtain superiorities against classical MIL methods on generalization ability, positioning ROIs, and comprehensibility.

2 Related work

In this section, we introduce some related works, including multiple instance learning, instance selection, attention mechanism, and gumbel softmax.

2.1 Multiple instance learning and instance selection

MIL [2, 5, 12–14] is a type of weakly supervised task, in which a labeled bag contains several unlabeled instances. The ROI [7,15] means the most positive instance in a positive bag. The standard assumption [1] in MIL refers to that a positive bag contains at least one positive instance, and the illustration can be found in Figure 1(a).

MIL methods contain two categories. Bag-level MILs [16–18] focus on aggregating instance representations into a single bag representation, in which some irrelevant instances may be disturbing. Instance-level MILs [6, 8, 19] devote to predicting instances individually, which can be used to identify ROIs, but their performances are usually worse as shown in [5]. Considering both of the two categories' ability, we propose a novel framework DMIS. In DMIS, we first figure key instances out, filtering the disturbing instances, and then make predictions using the selected instances.

Classical Instance selection [20] methods devote to filtering useless instances in a training set to reduce the sample complexity. In computer vision, instance selection methods such as AdaptIS [21] generates an object mask with the input of the whole image and a local point position. The instance selection process in MIL mainly focuses on mining key instances from a bag. Some traditional MIL methods [7, 15, 19, 22, 23] focus on detecting key instances or promoting bag-level classification accuracy via instance level

information. However, these methods cannot be directly applied to deep networks owing to the complex optimization strategies and the non-differentiable selection process. We apply a differentiable instance selection process to the deep MILs, which can be trained in an end-to-end mechanism.

2.2 Attention mechanism

Attention mechanism with deep learning was first applied to the neural machine translation task [24], and it has been used in MIL [11] as a permutation-invariant pooling operator to aggregate instance representations. The weights of instances are generated using a feed-forward fully-connected network, based on which the important instances can be determined. However, this is a post-selection mechanism and the continuous weights are not so comprehensible when compared with discrete codes.

As shown in [25], hard attention can obtain superiorities toward soft alignment, both on model performance and comprehensibility. However, hard attention is not so easy to train in an end-to-end framework, and some possible strategies based on reinforcement learning [25] or variational inference [26] are proposed. Recently, Ref. [27] introduces a new approach for hard attention in computer vision based on the magnitudes of the feature vectors and achieves a better performance. To obtain more precise ROIs and enhance the comprehensibility of MIL, we propose DMIS, utilizing hard attention to select key ROIs.

2.3 Gumbel softmax

In ROI analysis of MIL, we aim to sample the instance with the largest score, which corresponds to the standard assumption in MIL [1]. The classical approaches, such as the inverse transform sampling, can sample a one-hot code from the categorical distribution obtained by normalizing these scores. However, these sampling approaches are not differentiable and cannot be integrated into the neural network. Gumbel softmax [28, 29] is proposed to solve this problem, enabling the end-to-end training of discrete code in deep networks.

Under the framework of DMIS, we propose DMIS-GS based on gumbel softmax to generate categorical selection code for instance selection. As shown in discrete representation learning [30] and the binary activation gate in LSTM [31], categorical variables can be more precise and explainable than continuous ones. Considering the sampled one-hot codes may not be enough to completely figure ROIs out, we also utilize the gumbel top-k [32] to select several instances.

The temperature in gumbel softmax needs to be finely adjusted, and the normal method is an annealing process with decaying step by step. We find that normalizing the instance scores with the variance can make this hyper-parameter tuning process much easier.

3 Proposed methods

In this section, we will introduce the proposed framework DMIS and the method DMIS-GS.

3.1 Problem statements

In MIL, a dataset is denoted as $\boldsymbol{X} = \{(\boldsymbol{X}_i, y_i)\}_{i=1}^{N}$, containing N bags. Each bag $\boldsymbol{X}_i$ is represented as $\boldsymbol{X}_i = \{\boldsymbol{X}_{ij}\}_{j=1}^{N_i}$, containing N_i instances, and the bag-level label is $y_i \in \{0, 1\}$. The jth instance in the ith bag is a d-dimensional vector denoted as $\boldsymbol{X}_{ij}$. For simplicity, we sometimes omit the bag index and denote a bag as $\boldsymbol{X} = \{\boldsymbol{x}_i\}_{i=1}^{K}$ with K instances.

3.2 Hard selection vs. soft aggregation

In this subsection, we will show that in MIL with the standard assumption [1], hard selection can be much more appropriate for soft aggregation from both qualitative analysis and theoretical explanations.

Soft attention based MIL assigns a group of continuous weights to the instances, including weights for negative instances in the positive bag, which will result in an inaccurate aggregation for this positive bag. In some classification tasks, we expect that the sample representation should be far away from the oracle classification boundary as much as possible. If the aggregation process is inaccurate, the estimated positive bag representation will be more closer to the negative one, leading to a wrong decision, which is shown in Figure 1(b). However, hard selection can filter out the interference brought by negative instances, which is shown in Figure 1(c).

In order to give a theoretical explanation for hard selection, we make two assumptions as follows:

Assumption 1. First, both the positive and negative instances (embeddings) are distributed as clusters, for example, Gaussian distributions. Second, for a positive bag, the attention based MIL can always put more attention on the positive instances, and focus equally on negative instances.

The first assumption is a rational assumption in deep learning based methods, especially in deep metric learning [33, 34], where the between-class distances are maximized and the inner-class distances are minimized. The second one can introduce a simple attention weight formula which will be used in the following remark and corresponding theoretical analysis.

Remark 1 (Hard selection in standard MIL)**.** With the standard assumption considered in MIL, hard selection can be more appropriate compared with soft aggregation, both in promoting performance and in enhancing comprehensibility.

In attention based MIL, we denote the attention weights for $\boldsymbol{X}_{ij}$ as α_{ij} satisfying $\sum_{j=1}^{N_i}\alpha_{ij}=1$, and the true label for this instance as $\overline{y}_{ij}\in\{0,1\}$ (although inaccessible in training), while $y_i\in\{0,1\}$ denotes the obtainable bag label. We first aggregate instances in a bag as $\sum_{j=1}^{N_i}\alpha_{ij}\boldsymbol{X}_{ij}$, and then we empirically estimate the positive center as follows:

$$\hat{\boldsymbol{c}}^+=\frac{1}{\sum_{i=1}^{N}y_i}\sum_{i=1}^{N}y_i\left(\sum_{j=1}^{N_i}\alpha_{ij}\boldsymbol{X}_{ij}\right), \tag{1}$$

in which $\hat{\boldsymbol{c}}^+$ denotes the positive class center estimated from the aggregated bag-level representations. $\sum_{i=1}^{N}y_i$ is the number of positive bags.

It is obvious that we can get different estimated class centers with different attention weights α_{ij}, which will determine the quality of classifier boundary. We first give the oracle positive class center, which can be estimated from positive instances:

$$\hat{\boldsymbol{c}}_o^+=\frac{1}{\sum_{i=1}^{N}\sum_{j=1}^{N_i}\overline{y}_{ij}}\left(\sum_{i=1}^{N}\sum_{j=1}^{N_i}\overline{y}_{ij}\boldsymbol{X}_{ij}\right), \tag{2}$$

where $\hat{\boldsymbol{c}}_o^+$ refers to the oracle estimation for the positive class center using the instance labels $\overline{y}_{ij}$. The total number of positive instances is $\sum_{i=1}^{N}\sum_{j=1}^{N_i}\overline{y}_{ij}$.

We can also get the estimated negative class center using oracle instance labels as follows:

$$\hat{\boldsymbol{c}}_o^-=\frac{1}{\sum_{i=1}^{N}\sum_{j=1}^{N_i}(1-\overline{y}_{ij})}\left(\sum_{i=1}^{N}\sum_{j=1}^{N_i}(1-\overline{y}_{ij})\boldsymbol{X}_{ij}\right). \tag{3}$$

With the second assumption in Assumption 1, we define a simple α_{ij} for positive bags as follows:

$$\alpha_{ij}=\overline{y}_{ij}\frac{\mu}{N_i^+}+(1-\overline{y}_{ij})\frac{1-\mu}{N_i-N_i^+}, \tag{4}$$

in which μ is the total weight assigned to the N_i^+ positive instances in the ith positive bag. Then we analyze the decomposition of $\hat{\boldsymbol{c}}^+$ by replacing the α_{ij} in (1) with (4):

$$\begin{aligned}\hat{\boldsymbol{c}}^+ &= \frac{1}{\sum_{i=1}^{N}y_i}\sum_{i=1}^{N}y_i\left(\sum_{j=1}^{N_i}\frac{\mu}{N_i^+}\overline{y}_{ij}\boldsymbol{X}_{ij}+\sum_{j=1}^{N_i}\frac{1-\mu}{N_i-N_i^+}(1-\overline{y}_{ij})\boldsymbol{X}_{ij}\right)\\ &= \mu\cdot\frac{1}{\sum_{i=1}^{N}y_i}\sum_{i=1}^{N}\sum_{j=1}^{N_i}\frac{y_i}{N_i^+}\overline{y}_{ij}\boldsymbol{X}_{ij}+(1-\mu)\cdot\frac{1}{\sum_{i=1}^{N}y_i}\sum_{i=1}^{N}\sum_{j=1}^{N_i}\frac{y_i}{N_i-N_i^+}(1-\overline{y}_{ij})\boldsymbol{X}_{ij}\\ &\triangleq \mu\cdot\textsc{part}1+(1-\mu)\cdot\textsc{part}2,\end{aligned} \tag{5}$$

in which "PART1" and "PART2" are related to positive and negative centers (that is, (2) and (3)) respectively. If we assume N_i and N_i^+ in each positive bag nearly equal to one another, then we can get the following decomposition:

$$\hat{\boldsymbol{c}}^+\propto\lambda_1\mu\hat{\boldsymbol{c}}_o^+-\lambda_2(1-\mu)\hat{\boldsymbol{c}}_o^-, \tag{6}$$

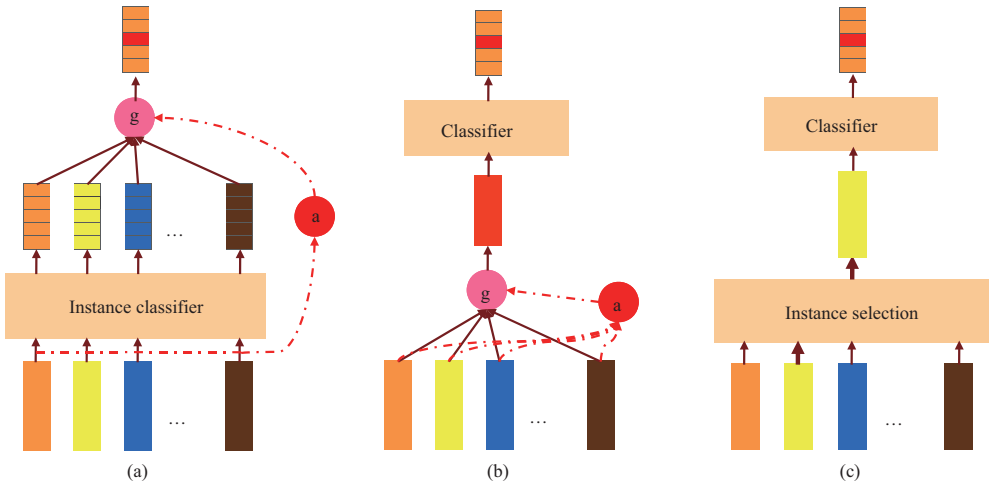

Figure 2 (Color online) Comparisons of our proposed DMIS with instance-level and bag-level frameworks in MIL. The "a" with a circle refers to the attention, and the dotted lines mean that the attention aggregation is optional. (a) Instance-level MILs first make predictions for instances, and then aggregate all predictions. (b) Bag-level MILs first aggregate instances together, and then make predictions. (c) Our proposed DMIS will first figure ROIs out, and then make predictions without the influence of disturbing instances.

in which λ_1 and λ_2 are constant values.

Following the aforementioned decomposition, we can find that a higher μ can make the estimated positive class center closer to the oracle one, which implies that in MIL, a distribution with smaller entropy (that is, from which hard selection codes can be sampled) will be more suitable for weighting instances. According to this phenomenon, we replace the soft attention based MIL with a hard one. Specifically, we can use a softer attention to explore instance relationships in the beginning, and ultimately we use a harder or even one-hot code to omit interference as much as possible.

3.3 Deep multiple instance selection (DMIS)

There are two classical frameworks for MIL, that is, the instance-level and bag-level MIL as categorized in [5, 11], which are illustrated in Figures 2(a) and (b). The two core components in MIL, namely the transformation and aggregation function, construct the general formulation for MIL:

$$F(\boldsymbol{X}) = f_{\boldsymbol{\theta}}\left(g_{\boldsymbol{\phi}}\left(f_{\boldsymbol{\psi}}\left(\boldsymbol{X}\right)\right)\right), \tag{7}$$

in which $f_{\boldsymbol{\theta}}(\cdot)$ and $f_{\boldsymbol{\psi}}(\cdot)$ are transformation functions and $g_{\boldsymbol{\phi}}(\cdot)$ is the aggregation function.

We propose a novel framework DMIS different from the two classical frameworks. First, we transform instances to a low-dimensional space via $f_{\boldsymbol{\psi}}(\cdot)$. Then we select key instances and aggregate them via $g_{\boldsymbol{\phi}}(\cdot)$, through which the disturbing instances can be filtered out. In the end, we apply a classifier $f_{\boldsymbol{\theta}}(\cdot)$ to make predictions for the bag. The whole framework is shown in Figure 2(c). It is worth mentioning that DMIS will first figure key instances out via the learned hard selection code and this process can be done in an end-to-end manner, which is the most different aspect from the bag and instance level methods.

Although we have demonstrated that in MIL with standard assumption, hard selection can be more appropriate than soft aggregation, there still exsit two core problems in utilizing hard selection for ROIs identification: how to optimize the instance selection process and how to precisely identify ROIs as much as possible. One possible way to solve the first problem is utilizing reinforcement learning (RL), such as REINFORCE for optimizing hard attention [25]. However, RL based methods need large amounts of episodes for exploration and it is more suitable for sequential decision making tasks. Hence, we select a neural reparametrization method, that is, DMIS-GS, for end-to-end training. We find DMIS-GS can inherently solve the second problem to some extent, which will be introduced in the next subsection.

3.4 DMIS-GS

We introduce the proposed DMIS-GS method for both simple and complex MILs in this subsection, which utilizes gumbel softmax and gumbel top-k respectively. Then we propose the variance normalization trick.

3.4.1 *For simple MIL*

DMIS-GS utilizes gumbel softmax [28, 29] to select instances in an end-to-end approach. For a bag of instances $\boldsymbol{X} = \{\boldsymbol{x}_i\}_{i=1}^K$, we apply the MIL framework as introduced in (7). For formula simplicity, we omit the instance mapping step, that is, $f_{\boldsymbol{\psi}}(\cdot)$ is an identity function.

In order to select the interested instances, we first obtain the importance scores for each instance via a pseudo score function. Also for formula simplicity, we use a linear score function with a sigmoid activation function as an example (while in practice, we can use a more complex score function):

$$s_i = \text{Sigmoid}\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b\right), \tag{8}$$

where $\boldsymbol{w}$ and b are the parameters of the score function. With the obtained pseudo scores $\boldsymbol{s} = (s_1, s_2, \ldots, s_K)$, we use the Gumbel-Max trick as a reparametrization process

$$u_i = \frac{\exp\left((\log s_i + g_i)/\lambda\right)}{\sum_{j=1}^K \exp\left((\log s_j + g_j)/\lambda\right)}, \tag{9}$$

where g_i is drawn from $\text{Gumbel}(0,1) = -\log(-\log(z))$, $z \sim \text{Uniform}(0,1)$, and $\boldsymbol{u} = (u_1, u_2, \ldots, u_K)$ is the obtained selection code. It is worth mentioning that the $\lambda \in (0, \infty)$ is the temperature deciding the approximation degree of the obtained $\boldsymbol{u}$ toward the one-hot code. A higher λ can make $\boldsymbol{u}$ smoother, which can be used for exploration in the beginning, whereas a smaller λ can lead to a harder $\boldsymbol{u}$.

With the obtained $\boldsymbol{u}$, we can get a one-hot code as follows:

$$j^\star = \arg\max_j \{u_j\}, \quad \boldsymbol{e} = \text{OneHot}(j^\star), \tag{10}$$

where $j^\star$ is the sampled index which satisfies $p\left(j^\star = k\right) \propto s_k$, and $\boldsymbol{e}$ is the one-hot code with the $j^\star$th element being 1 while others being 0. Then the selected instance is (here we denote $\boldsymbol{X} \in \mathcal{R}^{K\times d}$ as a matrix contains all instance representations)

$$g_{\boldsymbol{\phi}}(\boldsymbol{X}) = \boldsymbol{X}^{\mathrm{T}}\boldsymbol{e}. \tag{11}$$

With the selected instance $g_{\boldsymbol{\phi}}(\boldsymbol{X})$, we can apply a classifier $f_{\boldsymbol{\theta}}(\cdot)$ to get the bag-level predictions, and the posterior distribution of bag label y can be obtained

$$f_{\boldsymbol{\theta}}\left(g_{\boldsymbol{\phi}}(\boldsymbol{X})\right) = p_{\boldsymbol{\theta}}\left(y|g_{\boldsymbol{\phi}}(\boldsymbol{X})\right). \tag{12}$$

The optimization target for MIL dataset $\{(\boldsymbol{X}_i, y_i)\}_{i=1}^N$ with N bags is

$$\boldsymbol{\theta}^\star, \boldsymbol{\phi}^\star = \arg\max_{\boldsymbol{\theta},\boldsymbol{\phi}} \frac{1}{N}\sum_{i=1}^N \log p_{\boldsymbol{\theta}}\left(y_i|g_{\boldsymbol{\phi}}(\boldsymbol{X}_i)\right). \tag{13}$$

With the gumbel reparametrization trick, the whole framework can be trained end-to-end via gradient backpropagation. The gradients can be propagated via straight-through gradient estimator as introduced in [35] for the discrete code in (10), and the score function parameters $\boldsymbol{\phi} = \{\boldsymbol{w}, b\}$ can be updated.

3.4.2 *For complex MIL*

The proposed DMIS-GS earlier can only select one instance from a bag, which is not competent for complex MIL problems. For example, the collective assumption [5] in MIL assumes that only the definite combinations of instances can decide which class the bag belongs to. For example, only an image contains both sea and sand segments can be categorized as beach.

We enhance our methods with the ability of selecting k instances from bags to combine several instances together. As to DMIS-GS, we can use the gumbel top-k method as proposed in [32], which directly selects the instances with the top-k scores similar to (10). The process can be formed as follows:

$$j_1^\star, j_2^\star, \ldots, j_k^\star = \arg\text{top-}k_j\{u_j\}. \tag{14}$$

Then we can get k one-hot codes as $\boldsymbol{E} = (\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_k) \in \mathcal{R}^{K\times k}$. Finally, we can select these instances out and use a mean pooling operator to get instance combinations:

$$g_{\boldsymbol{\theta}}(\boldsymbol{X}) = \text{mean}\left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{E}, \text{dim} = 1\right). \tag{15}$$

Finally, we apply a classifier to make predictions for the whole bag as before, and we denote this proposed method as DMIS-GS-K, where K is the number of selected instances.

3.4.3 *Variance normalization*

As proposed in Subsection 3.3, one of the core problems for instance selection is how to figure out ROIs in a precise way as much as possible. At the beginning of the training, an exploration strategy should be used for searching key instances, and then the obtained key instances should be maximally exploited.

We propose that the temperature of DMIS-GS can inherently work for mining the key instances. A common method is using an annealing process for the temperature in (9). To be specific, the temperature is initialized as T_0 and then is decayed with a multiplier γ for every epoch. The lowest temperature is set as $T_{\min}$. These are important hyper-parameters for DMIS-GS. The hyper-parameters are not so easy to determine, and the best ones used in one task cannot be directly transferred to another task, which is due to the varying distributions of the obtained scores in (8). In our studies, we propose a variance normalization process as follows:

$$\overline{s}_i = \frac{1}{K}\sum_{k=1}^{K} s_i, \quad \hat{s}_i = \frac{s_i}{\sqrt{\sum_{i=1}^{K}\left(s_i - \overline{s}_i\right)^2/K}}. \tag{16}$$

The normalization process is similar to batch normalization [36] and instance normalization [37] methods in deep models, which can bring out more flat optimization regions as shown in [38]. With the normalized instance scores, we find that the hyper-parameters are much easier to determine and the best ones can be transferred among tasks, which is conformed to the explanation in [36] and will be verified in Subsection 4.2.3.

Therefore, the hyper-parameters matter a lot in our proposed DMIS-GS, and the variance normalization process can make it easier. We give Remark 2.

Remark 2 (Variance normalization)**.** A variance normalization process as shown in (16) can make the temperature tuning process much easier, that is, the determination of hyper-parameters $(T_0, \gamma, T_{\min})$ can be shared among tasks, leading to a better balance of exploration and exploitation in mining ROIs.

3.5 Theoretical analysis

As characterized in [39], the permutation invariant function family has a special structure which provides insight into designing deep network architectures that can operate on sets. The MIL problems can be considered as the set problems, in which the bags and instances are sets and elements correspondingly. We first display the theorem in [39], and then under which we can analyze our framework in a theoretical way and get a better understanding of our framework.

Theorem 1 (Permutation invariant functions [39])**.** A function $f(\boldsymbol{X})$ operating on a set $\boldsymbol{X}$ having elements from a countable universe, is a valid set function, that is, invariant to the permutation of instances in $\boldsymbol{X}$, iff it can be decomposed in the form $\rho(\sum_{\boldsymbol{x}\in\boldsymbol{X}}(\psi(\boldsymbol{x}))$, for suitable transformations $\rho(\cdot)$ and $\psi(\cdot)$.

The theorem implies that the set problem can be divided into three steps. First, a basic function $\psi(\cdot)$ is applied to each instance, and the output can be low-dimensional representations. Second, a summation operator is applied to aggregate all the representations. Third, another transformation function $\rho(\cdot)$ is applied to the aggregated representation. As further illustrated in [10], the summation operator is replaced with a maximization operator, and the max-pooling is utilized to aggregate instance embeddings in point cloud tasks. Both of them agree well with (7).

Being different from the aforementioned two set function formulas, our framework can be declared as Remark 3.

Remark 3 (Permutation invariant DMIS)**.** Our proposed DMIS can be formulated as $\rho(\sum_{\boldsymbol{x}\in\boldsymbol{X}}(\psi(\boldsymbol{x})))$, in which $\psi(\boldsymbol{x}) = \frac{1}{K}s(\boldsymbol{x})\cdot\boldsymbol{x}$. $s(\boldsymbol{x})$ is a selection function which returns a discrete value in $\{0,1\}$, and K is the number of selected instances.

Hence, our framework can also be applied to other set problems to obtain better performance and better comprehensibility, which shows that our framework is universal. However, we only focus on MIL problems in this paper.

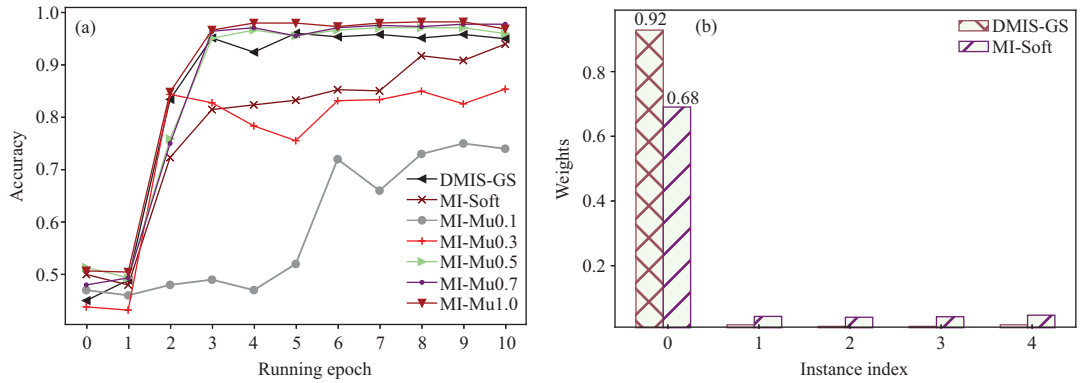

Figure 3 (Color online) Comparisons of classification performance and positioning ROIs with different aggregation weights in synthetic MIL dataset. (a) Classification accuracy. MI-Mu1.0 uses the oracle hard selection weights. DMIS-GS can get better performance than MI-Soft. (b) Comparison of positioning ROIs. DMIS-GS can identify ROIs more accurately than MI-Soft.

4 Experimental studies

In this section, we show that our proposed DMIS can obtain superiorities toward other MIL methods on both generalization ability, positioning ROIs, and comprehensibility using hard selection. We verify these through synthetic data, classical MIL datasets and sentiment classification datasets.

4.1 Synthetic data

4.1.1 *Details*

In order to demonstrate why hard selection facilitates MIL learning process, we construct a synthetic dataset with accessible instance labels. We can first show that soft weights for aggregation may inevitably introduce some interfering elements, and then we can compare hard selection and soft attention in the aspect of positioning ROIs quantitatively.

We construct a synthetic dataset as shown in Figure 1(a). First, we select two oracle gaussian distributions for positive and negative instances correspondingly, for example, $\boldsymbol{\mu}_1 = (1.0, 2.0)$, $\boldsymbol{\mu}_2 = (2.0, 1.0)$ and the $\boldsymbol{\Sigma}_1$, $\boldsymbol{\Sigma}_2$ are set to the diagonal matrix with definite value, for example, 0.1. Then we sample both positive and negative bags according to the standard assumption in MIL [1]. Specifically, we sample 5000 positive bags and 5000 negative bags as train set, 1000 positive bags and 1000 negative bags as the test set. For convenience, each bag contains 10 instances, and only the first instance is positive in the positive bag, while the instances in the negative bag are all negative.

We utilize a two-layer MLP with a hidden size of 32 to make predictions for the synthetic dataset. We use another two-layer MLP and sigmoid activation to generate instance scores for DMIS-GS. We can set the instance aggregation weights from an oracle view to make comparisons among different aggregation methods. Specifically, for the positive instance in the positive bag, we set its weight as $\mu \in \{0.1, 0.3, 0.5, 0.7, 1.0\}$, while the other negative instances with weight $(1-\mu)/(K-1)$. For the aforementioned synthetic data, we set instance aggregation weights with an oracle K-dim vector as $[\mu, (1-\mu)/(K-1), \ldots, (1-\mu)/(K-1)]$, and the model is referred as MI-Mu0.1 to MI-Mu1.0. We focus on comparing with MI-Soft [11], whose aggregation weights are learned in an end-to-end approach. We can use a larger batch size of 64 owing to that the synthetic data has the same number of instances in each bag. The optimizer is Adam with a learning rate of 0.001, and all of the networks use the same initialization.

As to the hyper-parameters in DMIS-GS, we initialize the temperature at $T_0 = 10.0$ at first and then multiply it by $\gamma = 0.98$ for every mini-batch. The minimal value of temperature is set as $T_{\min} = 0.1$. Also, we utilize the proposed variance normalization trick.

4.1.2 *Results and analysis*

In the aspect of model performance, we record the classification accuracy on the test set for every 15 mini-batches, and the plotted curves is shown in Figure 3(a). From this figure, we can observe that MI-Mu1.0 can get the best classification accuracy 0.998, which means that there is absolutely no disturbing information in the aggregation process of positive bags owing to the oracle selection code.

However, with the μ decreasing, such as MI-Mu0.7 and MI-Mu0.3, the classification performance drops a lot. At another extreme, the classification accuracy of MI-Mu0.1 is only about 0.75 when the weights in positive bags are uniform.

MI-Soft learns a group of soft aggregation weights, and can get a better performance than MI-Mu0.3, while worse than MI-Mu0.5. DMIS-GS can classify bags more accurately as shown in Figure 3(a), which tends to approach MI-Mu0.5 and MI-Mu0.7. We also find that DMIS-GS will almost always get a lower accuracy in the beginning training process with multiple trials, which is resulted from the exploration process with a higher temperature.

In order to determine whether DMIS-GS can focus on the ROIs more precisely or not when compared to soft aggregation, we investigate the identification of the positive instances in the 1000 test positive bags. We get the instance weights in each positive bag and calculate the mean values. For DMIS-GS, we first apply a softmax for the instance scores. The plotted mean weights of MI-Soft and DMIS-GS are shown in Figure 3(b), in which we only plot the weights of the first five instances for aesthetics. With the accessible instance labels in positive bags, we know that the first instance is positive while the others are negative ones. For MI-Soft, the mean weight assigned to the positive instance is about 0.68, while in DMIS-GS, the weight is almost 0.92, which verifies that DMIS-GS can locate ROIs more accurately. This also explains the performance in Figure 3(a), in which DMIS-GS can get classification accuracy higher than 0.9 in several batches while MI-Soft is much worse.

4.2 Classical MIL data

4.2.1 *Details*

In order to compare our methods with the classical MIL methods, we verify our proposed DMIS on classical MIL datasets. MUSK1 and MUSK2 [1] are drug activity prediction datasets, in which a bag is a molecule with multiple shapes. There are a total of 92 bags, 476 instances in MUSK1 and 102 bags, 6598 instances in MUSK2. TIGER, ELEPHANT and FOX [6] are the extracted image features, in which a bag is positive if only it contains the target animal features. There are total 1220, 1391, and 1320 instances in them correspondingly. There are 200 bags in each of them.

We compare our methods with both shallow and deep MILs. For shallow MILs, we select SVM based methods mi-SVM and MI-SVM [6], graph based method mi-Graph [40], descriptor based methods miVLAD and miFV [41]. For deep MILs, we compare with MI-Net and its variants MI-Net-DS, MI-Net-RC [42]. Specially, different aggregation methods, such as mean aggregation (that is, MI-Mean), max aggregation (that is, MI-Max), and attention aggregation (that is, MI-Soft [11]) are also compared.

We select the network and hyper-parameters as reported by MI-Net [42], and all of our experiments are repeated 10 times. The average results are reported. For DMIS-GS, we utilize the proposed variance normalization trick, and we initialize the temperature as $T_0 = 5.0$ at first and then multiply it by $\gamma = 0.9$ for every epoch, and the minimal value of temperature is set as $T_{\min} = 0.1$. And also, a two-layer MLP with sigmoid activation is utilized to generate instance scores.

4.2.2 *Results and analysis*

The classification accuracies can be found in Table 1. Our methods can obtain superiorities toward both classical methods and network-based MILs on image feature datasets, while a bit lower than classical methods on drug datasets. However, our methods can always get higher accuracies than MI-Soft in all datasets, which verifies the finding that hard selection can be more appropriate in MIL again.

We also show the ability to position ROIs in DMIS-GS. Because the instance labels are not obtainable, we use the entropy of instance aggregation weights as an alternative criterion. For DMIS-GS, we apply a softmax to the score as in (8) and calculate the mean entropy of learned aggregation weights. For MI-Soft, the attention weights are already processed via a softmax operator, and so that we can calculate the entropy directly. We select MUSK1 and ELEPHANT as examples and the changes of entropy on the test set in the learning process are plotted in Figure 4. We find that DMIS-GS tends to assign smoother weights for exploration at first and turns to harder ones sharply, while the entropy obtained by MI-Soft drops slower. Through this finding, we can know that DMIS-GS will explore a lot in the beginning, and then can locate ROIs quickly with a sharper instance selection distribution. Notably, the entropy is calculated on the test set and the instance weights of DMIS-GS are obtained directly from the scores

Table 1 Comparison results on five classical MIL datasets, and the classification accuracy is reported. "K" in "DMIS-GS-K" means the number of selected instances. DMIS-GS and DMIS-GS-K can get better results than most compared methods

	MUSK1	MUSK2	FOX	TIGER	ELEPHANT
mi-SVM	87.4	83.6	58.2	78.4	82.2
MI-SVM	77.9	84.3	57.8	84.0	84.3
mi-Graph	88.9	90.3	62.0	86.0	86.9
miVLAD	87.1	87.2	62.0	81.1	85.0
miFV	**90.9**	88.4	62.1	81.3	85.2
MI-Net	88.7	85.9	62.2	83.0	86.2
MI-Net-DS	85.9	87.4	63.0	84.5	87.2
MI-Net-RC	89.8	87.3	61.9	83.6	85.7
MI-Mean	89.1	89.6	60.2	83.5	86.9
MI-Max	87.6	84.7	58.1	81.2	85.7
MI-Soft	89.2	85.8	61.5	83.9	86.8
DMIS-GS	90.4	90.2	62.7	**86.6**	**87.9**
DMIS-GS-2	90.2	90.2	**63.9**	85.7	86.0
DMIS-GS-3	90.3	**90.7**	62.8	85.9	87.6

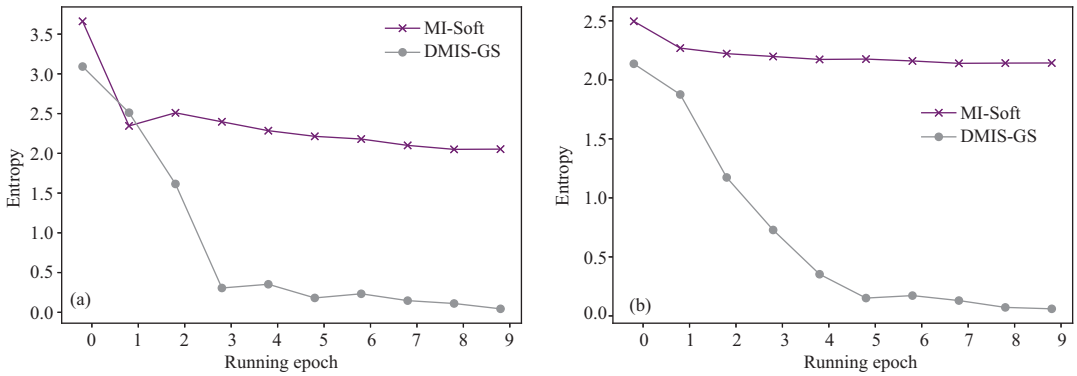

Figure 4 (Color online) The change of weight entropy on the test sets of MUSK1 (a) and ELEPHANT (b) in the learning process. DMIS-GS can obtain much lower entropy in the learning process.

before reparametrization, showing that the balance of exploration and exploitation also generalizes to test data.

4.2.3 *Ablation studies*

The temperature matters a lot in DMIS-GS as illustrated in Subsection 3.4.3. Hence, we investigate the influence of these hyper-parameters including the initial temperature T_0, the decaying multiplier γ, and the minimal temperature $T_{\min}$.

We guess that the instance scores in different datasets may have uneven distributions, which makes the adjusting process harder and cannot be shared. The following will present an experimental verification. First, we record instance scores (before sigmoid) of all bags without variance normalization in different datasets and plot their distributions as shown in Figure 5(a), It is obvious that the scores vary a lot across datasets, leading to the hardness of hyper-parameter selection. To be specific, the ELEPHANT instance scores vary in $[-1.64, 3.79]$, the FOX scores lie in $[-1.01, 0.96]$, while the TIGER scores vary in $[-0.95, 1.23]$. And also, the magnitude and skewness vary a lot too. However, with the normalization process applied, the score distributions become more similar across datasets as shown in Figure 5(b), which will make the knowledge of hyper-parameter selection more transferable.

Then, in order to show this in detail, we compare DMIS-GS with different settings of $(T_0, \gamma, T_{\min})$, and the compared results are listed in Table 2, in which the first column shows whether use the variance normalization or not.

From the listed results, we find that the performances on different tasks tend to be consistent with the variance normalization. For example, as shown in the left part of Table 2, the best hyper-parameters in all of the three datasets contain $\gamma = 0.9$ and $T_{\min} = 0.1$, and a much faster decaying method with a smaller $\gamma = 0.5$ will make the performance drops a lot. However, in the right part of Table 2, there

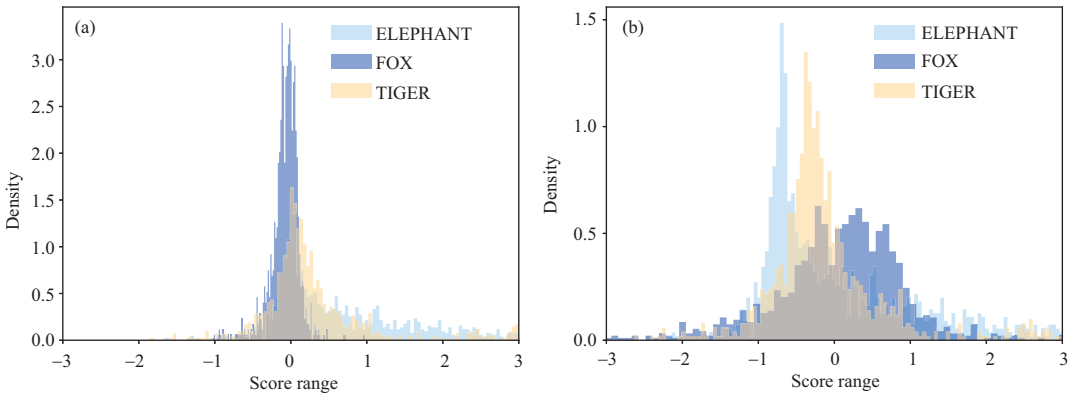

Figure 5 (Color online) The comparisons of instance score distributions before (a) and after (b) variance normalization. With variance normalization, the score distributions are more similar.

Table 2 Comparison results on different hyper-parameters of the temperature in DMIS-GS. With variance normalization, the best hyper-parameters in different datasets tend to have the same values

	With VarNorm			Without VarNorm		
$(T_0, \gamma, T_{\min})$	FOX	TIGER	ELEPHANT	FOX	TIGER	ELEPHANT
(5.0, 0.9, 0.1)	62.7	**86.6**	**87.9**	60.3	84.5	85.9
(1.0, 0.9, 0.1)	60.3	85.5	83.2	60.8	84.7	**86.3**
(10.0, 0.9, 0.1)	**62.1**	86.2	87.7	61.2	85.5	83.2
(5.0, 0.5, 0.1)	54.3	79.7	82.2	59.8	**85.8**	84.7
(5.0, 0.95, 0.1)	61.4	85.3	85.8	61.5	83.6	85.4
(5.0, 0.9, 0.5)	61.5	84.5	86.5	60.5	84.1	85.2
(5.0, 0.9, 0.01)	58.7	83.2	83.8	**61.9**	84.8	84.5

is basically nothing in common among the best hyper-parameters without the normalization process, implying that we need to search the best hyper-parameters for each dataset from scratch. The above experiments verify that the variance normalization trick could make the hyper-parameter determination process much more general across tasks.

4.2.4 *ROIs detection*

In order to show the ability of detecting ROIs, we apply DMIS-GS to some classical MIL datasets with accessible instance labels. We select Reuters text classification datasets used in [40] and report the precision in Table 3. The compared methods are mi-SVM [6], KI-SVM [7], VF, and VFr [23]. Note that our proposed DMIS-GS is mainly devoted to boosting the performance of MIL classification via selecting important instances, and it does not search out all key instances, so we only focus on the detection precision. From this table, we can find that our DMIS-GS can get a better ROIs detection performance, which will make the bag classification more accurate.

4.3 Sentiment classification data

4.3.1 *Details*

Sentiment classification [43, 44] aims to mine the emotions from texts. Especially with the rapid development of online business, mining the opinions and emotions from consumer reviews can facilitate consumers to quickly have a comprehensive understanding of specific products or services. Usually, the whole review (that is, the bag) contains multiple text segments (that is, instances) with varying emotions, which means that the MIL framework can be applied for sentiment analysis [3].

In this experiment, we evaluate our framework on three sentiment classification datasets, including Yelp13, Yelp14, and IMDB, in which the goal is to predict score ranks of product reviews. We declare that our framework can get higher or comparable results than other methods while keeping a comprehensive decision process. We use corresponding datasets which are smaller than the datasets reported in [43,44], and the preprocessing steps can be found in [43]. The descriptions of these datasets are listed as follows:

Table 3 The precision of detecting ROIs on some Reuters text datasets

	mi-SVM	KI-SVM	VF	VFr	DMIS-GS
alt.atheism	0.53	0.37	0.73	0.58	**0.78**
comp.graphics	0.61	0.38	0.66	**0.95**	0.93
comp.os.ms-windows.misc	0.55	0.39	0.79	0.55	**0.86**
comp.sys.ibm.pc	0.62	0.39	0.85	0.68	**0.86**
comp.sys.mac.hardware	0.78	0.32	0.78	0.70	**0.81**
comp.windows.x	0.55	0.40	0.69	0.27	**0.72**
misc.forsale	0.59	0.03	0.66	**0.85**	0.73
rec.autos	0.43	0.39	0.78	0.79	**0.82**
rec.motorcycles	0.40	0.71	0.70	0.42	**0.74**
rec.sport.baseball	0.46	0.63	0.73	**0.84**	0.79
rec.sport.hockey	0.45	0.83	0.79	**0.83**	0.82
sci.crypt	0.63	0.36	0.79	0.40	**0.85**
sci.electronics	0.95	0.39	0.96	0.90	**0.96**
sci.med	0.56	0.57	0.73	0.54	**0.83**
sci.space	0.37	0.30	0.86	0.76	**0.91**
soc.religion.christian	0.34	0.39	**0.84**	0.64	0.80
talk.politics.guns	0.52	0.36	0.53	0.66	**0.73**
talk.politics.mideast	**0.73**	0.66	0.72	0.58	0.71
talk.politics.misc	0.65	0.54	**0.72**	0.65	0.66
talk.religion.misc	0.30	0.38	0.67	0.49	**0.70**

Table 4 Comparison results on sentiment classification. ACC is classification accuracy, the higher the better, and MSE is the mean squared error, the lower the better. "K" in "DMIS-GS-K" means the number of selected instances

	Yelp13		Yelp14		IMDB	
	ACC	MSE	ACC	MSE	ACC	MSE
MILNET	63.8	0.477	63.7	0.465	45.8	2.12
HN-Mean	63.7	0.475	63.5	0.473	46.4	2.16
HN-Max	63.2	0.502	63.6	0.463	46.6	2.14
HAN	64.0	0.470	64.0	**0.455**	46.5	1.95
DMIS-GS	63.8	0.471	63.9	0.473	46.6	1.88
DMIS-GS-2	**64.8**	0.462	63.8	0.460	**46.9**	**1.86**
DMIS-GS-3	64.0	**0.460**	**64.5**	0.465	46.7	1.88

• **Yelp13** is the restaurant review dataset from Yelp Dataset Challenge in 2013, containing five levels of ratings and 78966 reviews in total.

• **Yelp14** is the restaurant review dataset from Yelp Dataset Challenge in 2014, containing five levels of ratings and 231163 reviews in total.

• **IMDB** is the movie review dataset, containing ten levels of ratings and 84919 reviews in total.

We apply DMIS-GS to the HAN [44] framework, selecting interested words or sentences in a hierarchical manner. To be specific, we use glove embeddings [45] and then feed the words representations into the network. First, at the sentence level, interested words are selected via DMIS-GS and the sentence representations can be obtained. Second, the sentence representations are fed into the document-level processing networks and similarly DMIS-GS is utilized for selecting interested sentences. Last, the document classification can be made.

We compare our methods to instance-level MILNET [3], HAN [44], variations of HAN with mean-pooling (HN-Mean) or max-pooling (HN-Max). The specific architecture and parameter settings can be found in HAN [44]. For DMIS-GS, we utilize the proposed variance normalization trick, and we initialize the temperature as $T_0 = 20.0$ at first and then multiply it by $\gamma = 0.95$ for every epoch, and the minimal value of temperature is set as $T_{\min} = 0.1$. And also, a two-layer MLP with sigmoid activation is utilized to generate scores for both words and sentences.

4.3.2 *Results and analysis*

Sentiment classification is much more complex for DMIS-GS, while it can still obtain comparable or a bit better performance compared with other methods as shown in Table 4. The best results are obtained via

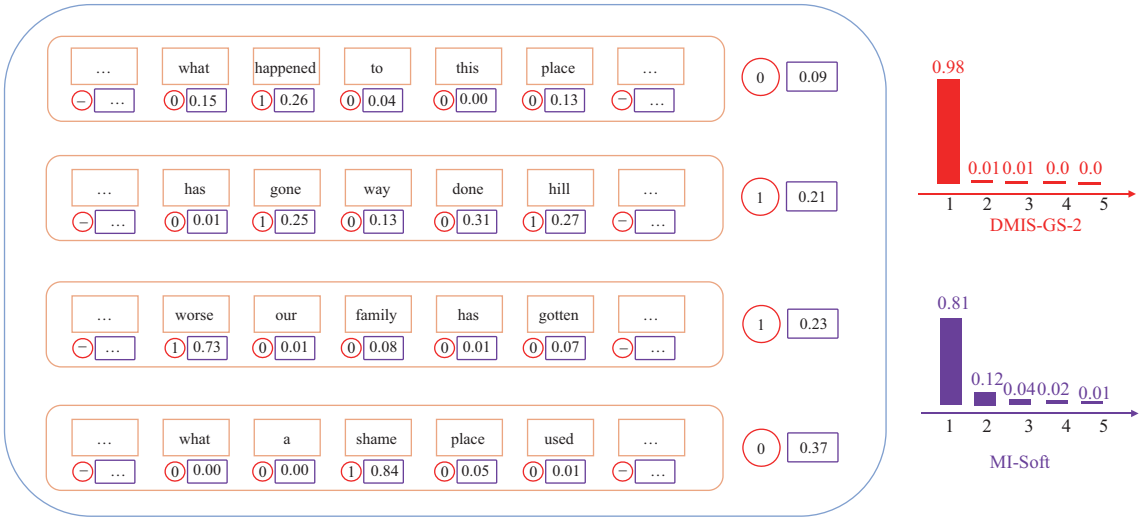

Figure 6 (Color online) Comparisons between the decision process of DMIS-GS-2 and HAN (MI-Soft). The red circle shows the selection of words or sentences in DMIS-GS-2, and the purple rectangle shows the weights learned from MI-Soft. DMIS-GS-2 can provide a clearer explanation for the final decision.

DMIS-GS-2 or DMIS-GS-3, which shows that some instance combinations should be considered in this task.

We further study the decision process of DMIS-GS-2 compared with HAN. It is notable that DMIS-GS-2 utilizes a hard selection code while HAN uses soft weights. For example, a randomly selected negative review (the score rank is one) is classified in the process shown in Figure 6, in which both the continuous codes of HAN (that is, MI-Soft) and discrete codes of DMIS-GS-2 are displayed. Although both models classify the review correctly (the plotted bars in Figure 6), it is obvious that DMIS-GS can provide discrete selection codes which makes the predictions much more comprehensible. From the decision process, we can locate the most emotional words or sentences quickly such as the displayed word "worse" and "shame".

5 Conclusion

We enabled deep MIL to figure ROIs out in an end-to-end approach automatically. In particular, we proposed a novel deep MIL framework DMIS, making predictions based on the selected instances, which is different from both of the bag and instance level MILs. We proposed DMIS-GS with gumbel softmax or gumbel top-k to better explore ROIs for optimizing the hard selection process, making the identified ROIs more accurate. In order to capture the knowledge across datasets better and make the hyper-parameter tuning process much easier, we proposed a variance normalization method in the adjusting process of the temperature. Some theoretical results were provided to show that our framework is general. Both theoretical analysis and abundant experiments verified that our proposed DMIS could get superiorities toward other MIL methods, both on model performance, positioning ROIs and comprehensibility. From another point of view, we also presented some cases (that is, MILs with standard assumption) in which hard attention can be more suitable than soft attention. Future work should focus on how to set the number of selected instances adaptively.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant Nos. 61773198, 61751306) and NSFC-NRF Joint Research Project (Grant No. 61861146001).

References

1 Dietterich T G, Lathrop R H, Lozano-Pérez T. Solving the multiple instance problem with axis-parallel rectangles. Artif Intell, 1997, 89: 31–71

2 Zhou Z H, Zhang M L, Huang S J, et al. Multi-instance multi-label learning. Artif Intell, 2012, 176: 2291–2320

3 Angelidis S, Lapata M. Multiple instance learning networks for fine-grained sentiment analysis. Trans Assoc Comput Linguist, 2018, 6: 17–31

4 Feng J, Zhou Z H. Deep MIML network. In: Proceedings of the 31st AAAI Conference on Artificial Intelligence, 2017. 1884–1890

5 Carbonneau M A, Cheplygina V, Granger E, et al. Multiple instance learning: a survey of problem characteristics and applications. Pattern Recogn, 2018, 77: 329–353

6 Andrews S, Tsochantaridis I, Hofmann T. Support vector machines for multiple-instance learning. In: Proceedings of Advances in Neural Information Processing Systems, 2002. 561–568

7 Li Y F, Kwok J T, Tsang I W, et al. A convex method for locating regions of interest with multi-instance learning. In: Proceedings of Machine Learning and Knowledge Discovery in Databases, European Conference, 2009. 15–30

8 Carbonneau M A, Granger E, Raymond A J, et al. Robust multiple-instance learning ensembles using random subspace instance selection. Pattern Recogn, 2016, 58: 83–99

9 Zhang Q, Goldman S A. EM-DD: an improved multiple-instance learning technique. In: Proceedings of Advances in Neural Information Processing Systems, 2001. 1073–1080

10 Qi C R, Su H, Mo K, et al. Pointnet: deep learning on point sets for 3D classification and segmentation. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2017. 77–85

11 Ilse M, Tomczak J M, Welling M. Attention-based deep multiple instance learning. In: Proceedings of the 35th International Conference on Machine Learning, 2018. 2132–2141

12 Tang P, Wang X G, Bai S, et al. PCL: proposal cluster learning for weakly supervised object detection. IEEE Trans Pattern Anal Mach Intell, 2020, 42: 176–191

13 Wang X G, Yan Y L, Tang P, et al. Bag similarity network for deep multi-instance learning. Inf Sci, 2019, 504: 578–588

14 Wei X S, Ye H J, Mu X, et al. Multi-instance learning with emerging novel class. IEEE Trans Knowl Data Eng, 2019. doi: 10.1109/TKDE.2019.2952588

15 Zhou Z H, Xue X B, Jiang Y. Locating regions of interest in CBIR with multi-instance learning techniques. In: Proceedings of the 18th Australian Joint Conference on Artificial Intelligence, 2005. 92–101

16 Chen Y X, Bi J B, Wang J Z. MILES: multiple-instance learning via embedded instance selection. IEEE Trans Pattern Anal Mach Intell, 2006, 28: 1931–1947

17 Wang J, Zucker J D. Solving the multiple-instance problem: a lazy learning approach. In: Proceedings of the 17th International Conference on Machine Learning, 2000. 1119–1126

18 Zhou Z H, Zhang M L. Solving multi-instance problems with classifier ensemble based on constructive clustering. Knowl Inf Syst, 2007, 11: 155–170

19 Viola P A, Platt J C, Zhang C. Multiple instance boosting for object detection. In: Proceedings of Advances in Neural Information Processing Systems, 2005. 1417–1424

20 Olvera-López J A, Carrasco-Ochoa J A, Martínez-Trinidad J F, et al. A review of instance selection methods. Artif Intell Rev, 2010, 34: 133–143

21 Sofiiuk K, Barinova O, Konushin A. Adaptis: adaptive instance selection network. In: Proceedings of IEEE/CVF International Conference on Computer Vision, 2019. 7354–7362

22 Li Z, Geng G H, Feng J, et al. Multiple instance learning based on positive instance selection and bag structure construction. Pattern Recogn Lett, 2014, 40: 19–26

23 Liu G Q, Wu J X, Zhou Z H. Key instance detection in multi-instance learning. In: Proceedings of the 4th Asian Conference on Machine Learning, 2012. 253–268

24 Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate. In: Proceedings of the 3rd International Conference on Learning Representations, 2015

25 Xu K, Ba J, Kiros R, et al. Show, attend and tell: neural image caption generation with visual attention. In: Proceedings of the 32nd International Conference on Machine Learning, 2015. 2048–2057

26 Deng Y T, Kim Y, Chiu J, et al. Latent alignment and variational attention. In: Proceedings of Advances in Neural Information Processing Systems, 2018. 9735–9747

27 Malinowski M, Doersch C, Santoro A, et al. Learning visual question answering by bootstrapping hard attention. In: Proceedings of the 15th European Conference on Computer Vision, 2018. 3–20

28 Jang E, Gu S, Poole B. Categorical reparameterization with gumbel-softmax. In: Proceedings of the 5th International Conference on Learning Representations, 2017

29 Maddison C J, Mnih A, Teh Y W. The concrete distribution: a continuous relaxation of discrete random variables. In: Proceedings of the 5th International Conference on Learning Representations, 2017

30 van den Oord A, Vinyals O, Kavukcuoglu K. Neural discrete representation learning. In: Proceedings of Advances in Neural Information Processing Systems, 2017. 6306–6315

31 Li Z H, He D, Tian F, et al. Towards binary-valued gates for robust LSTM training. In: Proceedings of the 35th International Conference on Machine Learning, 2018. 3001–3010

32 Kool W, van Hoof H, Welling M. Stochastic beams and where to find them: the gumbel-top-k trick for sampling sequences without replacement. In: Proceedings of the 36th International Conference on Machine Learning, 2019. 3499–3508

33 Do T T, Tran T, Reid I D, et al. A theoretically sound upper bound on the triplet loss for improving the efficiency of deep distance metric learning. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, 2019. 10404–10413

34 Qian Q, Shang L, Sun B G, et al. Softtriple loss: deep metric learning without triplet sampling. In: Proceedings of IEEE/CVF International Conference on Computer Vision, 2019. 6449–6457

35 Bengio Y, L'eonard N, Courville A C. Estimating or propagating gradients through stochastic neurons for conditional computation. 2013. ArXiv:1308.3432

36 Ioffe S, Szegedy C. Batch normalization: accelerating deep network training by reducing internal covariate shift. In: Proceedings of the 32nd International Conference on Machine Learning, 2015. 448–456

37 Ulyanov D, Vedaldi A, Lempitsky V S. Instance normalization: the missing ingredient for fast stylization. 2016. ArXiv:1607.08022

38 Santurkar S, Tsipras D, Ilyas A, et al. How does batch normalization help optimization? In: Proceedings of Advances in Neural Information Processing Systems, 2018. 2488–2498

39 Zaheer M, Kottur S, Ravanbakhsh S, et al. Deep sets. In: Proceedings of Advances in Neural Information Processing Systems, 2017. 3391–3401

40 Zhou Z H, Sun Y Y, Li Y F. Multi-instance learning by treating instances as non-i.i.d. samples. In: Proceedings of the 26th International Conference on Machine Learning, 2009. 1249–1256

41 Wei X S, Wu J X, Zhou Z H. Scalable algorithms for multi-instance learning. IEEE Trans Neural Netw Learn Syst, 2017, 28: 975–987

42 Wang X G, Yan Y L, Tang P, et al. Revisiting multiple instance neural networks. Pattern Recogn, 2018, 74: 15–24

43 Tang D Y, Qin B, Liu T. Document modeling with gated recurrent neural network for sentiment classification. In: Proceedings of Conference on Empirical Methods in Natural Language Processing, 2015. 1422–1432

44 Yang Z C, Yang D Y, Dyer C, et al. Hierarchical attention networks for document classification. In: Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2016. 1480–1489

45 Pennington J, Socher R, Manning C D. Glove: global vectors for word representation. In: Proceedings of Conference on Empirical Methods in Natural Language Processing, 2014. 1532–1543