Expert Systems With Applications 189 (2022) 116117

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Image classification with multi-view multi-instance metric learning

Jingjing Tang [a,b], Dewei Li [c], Yingjie Tian [c,d,e,*]

[a] School of Business Administration, Faculty of Business Administration, Southwestern University of Finance and Economics, Chengdu 611130, China
[b] Institute of Big Data, Southwestern University of Finance and Economics, Chengdu 611130, China
[c] Research Center on Fictitious Economy and Data Science, Chinese Academy of Sciences, Beijing 100190, China
[d] School of Economics and Management, University of Chinese Academy of Sciences, Beijing 100190, China
[e] Key Laboratory of Big Data Mining and Knowledge Management, Chinese Academy of Sciences, Beijing 100190, China

ARTICLE INFO

Keywords:
Metric learning
Multi-view learning
Multi-instance learning
Image classification

ABSTRACT

Image classification is a critical and meaningful task in image retrieval, recognition and object detection. In this paper, three-side efforts are taken to accomplish this task. First, visual features with multi-instance representation are extracted to characterize the image due to the merits of bag-of-words representations. And a new distance function is designed for bags, which measures the relationship between bags more precisely. Second, the idea of multi-view learning is implemented since multiple views encourage the classifier to be more consistent and accurate. Last but not least, the metric learning technique is explored by optimizing the joint conditional probability to pursue view-dependent metrics and the importance weights of the newly-designed distance in multi-view scenario. Therefore, we propose a multi-view multi-instance metric learning method named MVMIML for image classification, which integrates the advantages of the multi-view multi-instance representation and metric learning into a unified framework. To solve MVMIML, we adopt the alternate iteration optimization algorithm and analyze the corresponding computational complexity. Numerical experiments verify the advantages of the new distance function and the effectiveness of MVMIML.

1. Introduction

With explosive growth of image data from daily life and the Internet, image classification and recognition have been an active research spot for many years. The far-reaching study can be employed in practical applications, including face recognition (Qiu et al., 2021), species categorization (Pang et al., 2021), object detection (Wang et al., 2021), and so forth. However, there exist three challenges in this research. The first challenge is to extract numerical features from images since the original image cannot be exploited in the traditional machine learning methods directly. A series of existing studies have focused on this area. Classical methods for the feature extraction include haar-like feature (HAAR) (Lienhart & Maydt, 2002), scale-invariant feature transform (SIFT) (Lowe, 2004), histograms of oriented gradients (HOG) (Dalal & Triggs, 2005), local binary pattern (LBP) (Ahonen et al., 2004), speeded up robust feature (SURF) (Bay et al., 2006) and so on. The above methods can be classified into two categories: single feature vector representation (HAAR, HOG, LBP) and bag-of-words representation (SIFT, SURF, patches of LBP and HOG). The difference between these two kinds of features is whether an image is represented by a single vector or a bag of instances, leading to two areas in machine learning: standard classification and multi-instance learning. Since the contents in an image are not distributed uniformly or regularly, bag-of-words representation has the advantage that each word expresses an image's key feature independently (González et al., 2017). As a result, the negative impact of unfixed locations of these key features can be mitigated in a single vector.

Real-world data are usually collected from diverse domains or obtained from various feature extractors. These data can be naturally partitioned into distinct feature sets, each of which is regarded as a particular view (Tang et al., 2021a, 2019, 2017). The second challenge is to integrate the information from multiple feature sets effectively. Multi-view learning (MVL) focuses on learning with such multiple views for performance improvement. To guarantee its success, most MVL algorithms concentrate on either view consistency or view diversity corresponding to the consensus principle or the complementarity principle. The former principle targets at maximizing the agreement among multi-view features, while the latter principle emphasizes the complementary information among views. Such two principles serve as an important guide in multi-view modeling. Existing multi-view learning algorithms can be categorized into three groups: (1) co-training, (2) multiple kernel learning, and (3) subspace learning. Extensive

[*] Correspondence to: No. 80 of Zhongguancun East Street, Haidian District, Beijing, 100190, PR China.
E-mail addresses: tjj@swufe.edu.cn (J. Tang), lidewei06@163.com (D. Li), tyj@ucas.ac.cn (Y. Tian).

https://doi.org/10.1016/j.eswa.2021.116117
Received 24 April 2021; Received in revised form 17 August 2021; Accepted 16 October 2021
Available online 24 October 2021

experiments have verified that leaning with multiple views contribute to boosting the performance.

Based on the idea that the desired metric should shrink the distance between similar points and expand the distance between dissimilar points as much as possible, elaborate efforts have been made on metric learning. For each image, we can extract its features from multiple views, and each view consists of multi-instance representation. When confronted with multi-view multi-instance features, how to measure the distance between images has been rarely studied. Thus, the third key challenge is to exploit an efficient data-dependent distance function to characterize the image relationships with such feature representation and the distributions in the feature space.

To address the above-mentioned challenges, it is worthy and important to fully explore the relationship among different views of one image or among distinct bags in the same view. Therefore, we develop multiple view-dependent metrics in multi-instance task with multi-view representation.

In this paper, a new multi-view multi-instance metric learning method, named MVMIML, is proposed for image classification. Due to the merits of bag-of-words representations and multiple views, we extract multi-instance multi-view features for each image. To leverage such features effectively, we define a new distance function for bags, and then pursue a view-dependent metric by maximizing the average conditional probability to guarantee that every image is similar to its nearest image. The distance between images is computed by the weighted sum of the distance between bags from each single view. Since the metrics and weights are both required to be optimized, we adopt the alternate optimization strategy to solve our proposed model. The comprehensive experiments verify the effectiveness of MVMIML for image classification.

The main contributions are summarized as follows:

- We propose a new multi-view multi-instance metric learning method (MVMIML), which integrates the merits of both the multi-view multi-instance representation and metric learning into a unified framework.
- A new distance function is designed for bags, which measures the relationship between bags more precisely. Numerical experiments have validated the effectiveness of the new distance function.
- Alternating iteration optimization algorithm is adopted to solve MVMIML and we further theoretically analyze the corresponding computational complexity.
- The extensive experiments on the image datasets confirm that MVMIML compares more favorably than other benchmark algorithms.

The remainder of this paper is organized as follows. In Section 2, we introduce the related works about metric learning, multi-view learning and multi-instance classification. Our model and the corresponding optimization are provided in Section 3. In Section 4, numerical experiments are performed to verify the effectiveness of our proposed method. Finally, we conclude the paper in Section 5.

2. Related works

2.1. Metric learning

Metric learning aims to learn a distance function to improve the performance of distance-related methods such as kNN and k-means methods. Consider a dataset with c classes

$$T = \{(x_1, y_1), \dots, (x_m, y_m)\}, \tag{1}$$

where $(x_i, y_i) \in R^n \times \{1, 2, \dots, c\}, i = 1, \dots, m$ and m is the total number of samples, and n is the number of features. Two sets are defined as follows

$$S = \{(x_i, x_j) | y_i = y_j\}, \tag{2}$$

$$D = \{(x_i, x_l) | y_i \neq y_l\}. \tag{3}$$

The points in each pair of S are from the same class and D contains pairs of dissimilar points. Metric learning seeks for a metric to recompute the distance between two different points as

$$d_M(x_i, x_j) = (x_i - x_j)^\top M (x_i - x_j), \tag{4}$$

to make similar points closer and dissimilar points farther. In (4), an effective metric M should satisfy the following conditions (Wang & Sun, 2014):

- distinguishability: $d_M(x_i, x_i) = 0$;
- non-negativity: $d_M(x_i, x_j) \geq 0$;
- symmetry: $d_M(x_i, x_j) = d_M(x_j, x_i)$;
- triangular inequality: $d_M(x_i, x_j) + d_M(x_i, x_k) \geq d_M(x_j, x_k)$.

Numerous metric learning methods have been proposed to show strong ability of view-dependent distance in adjusting the original structure of the feature space, so that more advantageous neighborhoods can be formed. The label information of pairwise relationship in (2)–(3) has been exploited in most previous works. One of the earliest efforts in pursuing ideal metric was metric learning with side information (MLSI) (Xing et al., 2002). It used similarity side-information to improve the performance of kNN based on the idea that similar points should be as near as possible and the distance between dissimilar points should be larger than a threshold. The method was solved by positive semi-definite programming with high time complexity and the performance was not significantly better than traditional kNN. Goldberger et al. proposed neighborhood component analysis approach (NCA) (Goldberger et al., 2004) which directly maximized leave-one-out accuracy by learning a low-rank quadratic metric. But the computational complexity was also very high due to the leave-one-out strategy. Based on NCA, large margin nearest neighbor method (LMNN) (Weinberger & Saul, 2009) was developed to minimize the distance between any two similar and close points with the constraints that the points associated with different labels should be pushed away from the its neighborhood. Due to the limitations of LMNN, several extensions were introduced to improve LMNN, including solving LMNN more efficiently (Park et al., 2011) and introducing kernels into LMNN (Torresani & Lee, 2006). Globerson and Roweis provided an algorithm for learning a quadratic Gaussian metric (Mahalanobis distance) in classification tasks (Globerson & Roweis, 2005). In Miao et al. (2015), Miao et al. proposed a locally adaptive weighted distance-metric learning method to deal with the non-linearity of the data. From the view of information theory, Davis et al. presented a information-theoretic metric learning model (ITML) (Davis et al., 2007) to minimize the relative entropy between two multivariate Gaussian distribution, leading to a Bregman optimization problem.

The above methods are all proposed only for standard classification or clustering. Metric learning for the special tasks has also been studied extensively. A semi-supervised multi-view distance metric learning (SSM-DML) was proposed to learn the multi-view distance metrics from multiple feature sets and from the labels of unlabeled cartoon characters simultaneously (Yu et al., 2012). Jin et al. (2009) developed an iterative metric learning algorithm for multi-instance multi-label problem to improve the quality of associations between instances and class labels. Multi-Instance metric Learning (MIMEL) (Xu et al., 2011) aimed to maximize inter-class bag distance and minimize intra-class bag distance by constructing a minimization problem of KL divergence between two multivariate Gaussians.

2.2. Multi-view learning

Many real-world applications involve data with multiple forms of representation or “views”. For instance, the identification of one person can be represented by voice, fingerprint, iris and facial structure. As an active research field in machine learning, multi-view learning leverages the information from multiple views for better performance (Cano, 2017). Existing multi-view learning algorithms can be divided into three categories: (1) co-training, (2) multiple kernel learning, and (3) subspace learning.

Co-training algorithms are semi-supervised learning methods, which make the maximum mutual agreement on two views iteratively for view consistency. In Wang and Zhou (2010), Wang and Zhou considered co-training as the combination of label propagation over two views and unified the graph- and disagreement-based semi-supervised learning into one framework. The co-training cross-view based graph random walk approach (Wang et al., 2017) focused on learning cross-view distance measure by exploiting multiple graphs structure of multi-view data.

Multiple kernel learning (MKL) algorithms utilize kernels corresponding to different views and combine them either linearly or non-linearly. For example, the SimpleMKL algorithm used a gradient descent wrapping algorithm based on the standard SVM solver to iteratively determine the combination of kernels for MKL (Rakotomamonjy et al., 2008). By integrating nonparallel support vector machine (NPSVM) into the MKL framework, Tang and Tian proposed a model termed as MKNPSVM to learn the optimal kernel combination (Tang & Tian, 2017).

Subspace learning algorithms assume that the input views come from a latent subspace and aim at achieving this latent subspace shared by multiple views. Canonical correlation analysis (CCA) (Hotelling, 1936) and its kernel version called kernel canonical correlation analysis (KCCA) (Akaho, 2006) were two early works in subspace learning. They pursued basic vectors for two sets of variables, each of which corresponded to a single view. Multi-view least squares support vector machines (MV-LSSVM) incorporated information from all views in the training phase while still allowed for some degree of freedom to model the views differently (Houthuys et al., 2018). Tang et al. proposed a simple yet effective coupling privileged kernel method for multi-view learning (Tang et al., 2019) and further extended it to transfer learning (Tang et al., 2021a). Based on the restricted Boltzmann machine (RBM) and CCA, correlation RBM computed multi-view representations by regularizing the marginal likelihood function with the consistency across multiple views (Nan Zhang & Jia, 2019). By inheriting the asymmetric merit of LINEX loss, Tang et al. presented a general multi-view LINEX SVM framework including two models called MVLSVM-CO and MVLSVM-SIM (Tang et al., 2021b).

2.3. Multi-instance learning

Different from standard supervised learning, in which the input is described by a single feature vector, every input in multi-instance learning (MIL) is a set of labeled instances called a bag. A bag is flagged as positive if at least one instance in that bag is positive; otherwise, the bag is labeled as negative. The instance level MIL methods attempt to find positive instances in each bag to achieve a bag level classifier by aggregating the instance level classifier (He et al., 2020). In Dietterich et al. (1997), Dietterich introduced the multi-instance problem in the study of drug activity prediction. The molecule and the isomers within a molecule in the MIL was named as a bag and instances respectively. If there was one effective isomer, then the molecule can be defined as active, otherwise it was inactive. They developed three Axis-Parallel Rectangles (APR) learning algorithms to find the best axis-parallel rectangles that covered the maximum positive instances with the lowest cost in the attribute space. To deal with the instances of which the labels are ambiguous, Xiao et al. presented a similarity-based multiple-instance learning approach (SMILE) by considering the similarity of ambiguous instances to the positive class and the negative class (Xiao et al., 2013).

The bag level and embedding level MIL methods transfer the bag into a vector through distance measure such as kernel distance measure. The former directly calculates the distance between any two bags in different formulations and generates the bag-level predictions by performing voting on the instance predictions. The latter learns the relationship among instances by projecting them into a new embedding space. For example, Citation-kNN (Wang & Zucker, 2000) was bag level MIL method which decided the label of a bag not only by its neighbors but also its cites. In Maron and Lozano-Pérez (1998), a probabilistic framework called Diverse Density (DD) was to learn a concept by maximizing a defined likelihood function. The Expectation-Minimization diversity density (EMDD) method tried to identify the instance with highest diversity density, which was assumed to be the positive instance in each bag as determined by the EM algorithm (Zhang & Goldman, 2001). Melki et al. presented a novel bag-level representative multi-instance learning SVM framework named MIRSVM (Melki et al., 2018).

To our knowledge, there are few researches on the multi-instance learning with multiple views using the technique of metric learning. MVMIML is a probability framework, which defines a distance function for multi-instance learning problem and integrates the information from multiple views by learning multiple metrics. The experiments verify that our approach is effective in dealing with image classification with multi-view multi-instance representations.

3. Model and optimization

3.1. Multi-view multi-instance learning task

Since the features of every image can be extracted in the form of bags, each of which contains multiple instances, we can regard image classification as a multi-instance learning task. To begin with, we give the formal description for multi-instance learning associated with the training set as follows.

$$T = \{(X_i, y_i)\}_{i=1}^{m}, \tag{5}$$

where $X_i = \{x_{i1}, \ldots, x_{im_i}\}$ is a bag including m_i instances, and $y_i \in \{1, \ldots, c\}$ is the corresponding label. Each instance x_{ik} in X_i is a n-dimensional real vector. If and only if at least one instance in a bag belongs to class c, the bag belongs to class c.

The attributes of every image can be extracted from multiple feature descriptors termed as multi-view data. Thus, by representing the image with v views, the training set (5) for kth ($k = 1, 2, \ldots, v$) view is rewritten as follows:

$$T^k = \{(X_i^k, y_i)\}_{i=1}^{m} \tag{6}$$

where $X_i^k = \{x_{i1}^k, x_{i2}^k, \ldots, x_{im_i^k}^k\}, y_i \in \{1, 2, \ldots, c\}$. The bag X_i^k contains m_i^k instances and each is a real vector with n^k dimensions. The number of bags in different views is the same, but each bag in different views contains different number of instances.

The goal is to find a prediction function f with respect to multiple distance metrics $M_1, \ldots, M_v$, and each corresponds to an unique view. Given an image with the information from v views, $X^1, X^2, \ldots, X^v$, the label of the image can be predicted by $y = f(X^1, X^2, \ldots, X^v; M_1, M_2, \ldots, M_v)$.

3.2. Distance between bags

In the traditional distance metric learning, the following formulation is used to measure the distance between two feature vectors x_i and x_j

$$d_M(x_i, x_j) = (x_i - x_j)^\top M(x_i - x_j), \tag{7}$$

where the metric M should satisfy the property of distinguishability, non-negativity, symmetry and triangular inequality. However, in the feature extraction of image, the feature is in the form of bag, which contains multiple vectors. It is not suitable to concatenate these vectors into a longer vector. So it is hard yet very important to measure the

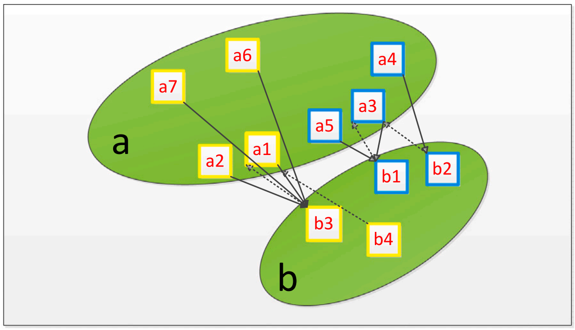


Fig. 1. A semantic explanation for D_{am}. The distinct bags **a** and **b** include several instances represented by the squares. For each instance in **a**, it finds the nearest instance in bag **b** and the distance is denoted by a solid line. For each instance in **b**, it finds the nearest instance in bag **a** and the distance is denoted by a dashed line. $D_{am}(a,b)$ equals to the average of all these distances.

distance of different images, each of which is represented by a bag of instances.

In metric learning, two distance functions are commonly utilized to measure the distances between bags as accurately as possible. The first one is the average distance of pairwise examples from different bags (Xu et al., 2011) formulated as

$$D_{ave}(X_i, X_j; M) = \frac{1}{m_i m_j} \sum_{k=1}^{m_i} \sum_{l=1}^{m_j} d_M(x_{ik}, x_{jl}), \tag{8}$$

where $X_i = \{x_{i1}, x_{i2}, \dots, x_{im_i}\}$ and $X_j = \{x_{j1}, x_{j2}, \dots, x_{jm_j}\}$ are two bags. The second one is the minimum distance of pairwise instances (Jin et al., 2009) called *minimal Hausdorff distance* defined as

$$D_{min}(X_i, X_j; M) = \min_{1 \le k \le m_i, 1 \le l \le m_j} d_M(x_{ik}, x_{jl}). \tag{9}$$

However, the two functions exist some shortages. For the function D_{ave} as (8), calculating the distance of pairwise instances can bring much redundant information, and the distance between two same bags is not zero for D_{ave}. For the function D_{min} as (9), it determines the distance of bags only by the minimum distance of pairwise instances, which ignores too much useful information. If two different bags contain similar instances, D_{min} will make improper judgment. Therefore, both D_{ave} and D_{min} cannot measure the bag distance properly.

Motivated by these, we design a new distance function. To begin with, an intuitional geometrics interpretation is illustrated in Fig. 1. For each instance x in each bag, we find the nearest instance in the other bag and record the corresponding distance. Such distance is the minimum for x in searching the other bag. We then calculate the average of these minimums, named as D_{am}. Formally, the newly-designed distance function D_{am} between two bags X_i and X_j is provided as follows

$$D_{am}(X_i, X_j; M) = \frac{1}{m_i} \sum_{p=1}^{m_i} \min_{x_{jl} \in X_j} d_M(x_{ip}, x_{jl}) + \frac{1}{m_j} \sum_{q=1}^{m_j} \min_{x_{ih} \in X_i} d_M(x_{jq}, x_{ih}). \tag{10}$$

We explain the advantage of the defined distance visually in Fig. 2. The verification of the effectiveness of our proposed distance function is performed in Section 4.

In multi-view scenario, every image I is represented by v views $X_i^1, X_i^2, \dots, X_i^v$. The distance between two images I and J is defined as

$$D_M(I, J) = \sum_{k=1}^{v} \alpha_k D_{am}(X_i^k, X_j^k; M_k), \tag{11}$$

where $\alpha_i,\ i = 1, \dots, v$ are the weights to be learned.

3.3. Metric learning in probability framework

Although the distance between bags is designed, the relationship between instances is also very important since it affects bag distance significantly. Fortunately, metric learning can be applied to establish favorable relationships for instances from bags. Inspired by the ideas of kNN and NCA (Goldberger et al., 2004), we aim to maximize the probability that one image's nearest image has the same label with it. In multi-view situation, we consider optimizing the joint conditional probability distribution of the image I with v views, but not simply maximize the sum of the marginal probability distribution, i.e.,

$$p(y_i | X_i^1, X_i^2, \dots, X_i^v; \mathscr{M}) = \frac{\exp(-f(X_i, y_i))}{\sum_{y=1}^{c} \exp(-f(X_i, y))}, \tag{12}$$

where

$$f(X_i, y) = \min_{y_j = y} D_M(I, J) = \min_{y_j = y} \sum_{k=1}^{v} \alpha_k D_{am}(X_i^k, X_j^k; M_k), \tag{13}$$

and $\mathscr{M} = \{M_1, \dots, M_v\}$.

On the whole training set, the following regularized likelihood function is constructed to learn the desired distance metrics

$$\begin{aligned} &\min_{\mathscr{M},\alpha} E(\mathscr{M}, \alpha) \\ &= -\frac{1}{m} \sum_{i=1}^{m} \ln p(y_i | X_i^1, X_i^2, \dots, X_i^v; \mathscr{M}) + \frac{\lambda}{2} \sum_{k=1}^{v} \|M_k\|_F^2 + \frac{\mu}{2} \|\alpha\|^2 \\ &= \frac{1}{m} \sum_{i=1}^{m} f(X_i, y_i) + \frac{1}{m} \sum_{i=1}^{m} \ln(\sum_{y=1}^{c} \exp(-f(X_i, y))) + \frac{\lambda}{2} \sum_{k=1}^{v} \|M_k\|_F^2 + \frac{\mu}{2} \|\alpha\|^2. \end{aligned} \tag{14}$$

To ensure the basic property of metric, $E(\mathscr{M}, \alpha)$ should satisfy $M_k \ge 0, k = 1, \dots, v$ (positive semi-definite).

Let $v = 1$, the primary model (14) degenerates into a single view version and the objective function becomes

$$\min_{M,\alpha} E_s(M, \alpha) = -\frac{1}{m} \sum_{i=1}^{m} \ln p(y_i | X_i; M) + \frac{\lambda}{2} \|M\|_F^2, \tag{15}$$

where $M \ge 0$ (positive semi-definite). The model (15) can be used to learn the metric in multi-instance classification with a single view. The workflow of the MVMIML method are summarized in Fig. Fig. 3.

3.4. Optimization

Since the models (14) and (15) are constructed with the constraints that all the metrics should be positive semi-definite, the optimal metrics can be achieved in the following strategy for simplicity. First, we can reduce the models (14) and (15) to unconstrained minimization problems. Due to the minimization of such problems, mini-batch stochastic gradient descent algorithm (Khalilpourazari et al., 2021) is adopted to obtain $M_1,\ M_2,\ \cdots,\ M_v$. Then we project such metrics $M_1,\ M_2, \dots,\ M_v$ into positive semi-definite space. In multi-view scenario, problem (14) involves complex optimization with respect to view-dependent metrics $\{M_k\}_{k=1}^{v}$ and importance weights $\{\alpha_k\}_{k=1}^{v}$ for the newly-designed distance. Similar to Tang et al. (2021b), we can decompose (14) into two sub-problems respectively in an alternating optimization procedure, where each sub-problem is solved by the mini-batch stochastic gradient descent algorithm. Note that sub-problem with respect to $\{M_k\}_{k=1}^{v}$ can be solved by above two-step strategy. Iterations are repeated until convergence or a maximum number of iterations is reached.

In single view scenario, only one metric M need to be optimized. The model (15) can be solved in the Algorithm 1. Given an unknown image L with bag X_l, its label can be decided by

$$y_l = \arg \max_{y=1,\dots,c} p(y | X_l; M^*). \tag{16}$$

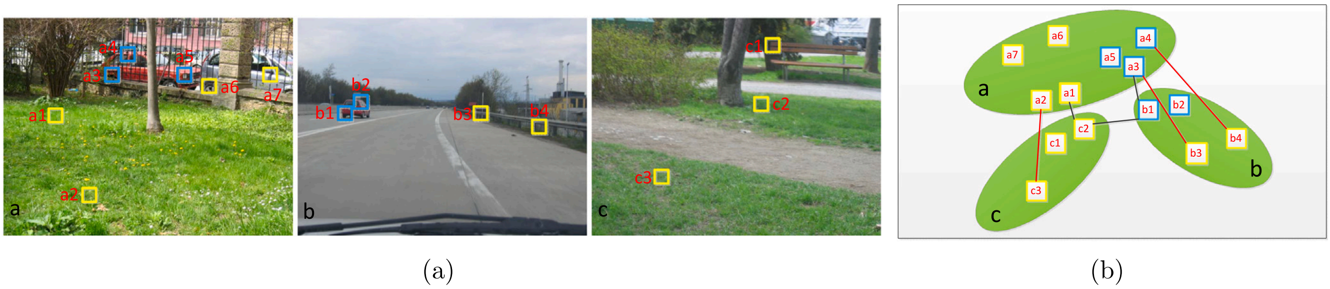


Fig. 2. A simple task to compare three kinds of bag distance D_{ave}, D_{min} and D_{am}. Given a picture **a** belonged to the class of *car*, find the picture with the same label from **b** and **c**. There are 7, 4 and 3 key points in the picture **a, b** and **c** respectively. The blue squares are the key points indicating the label. On the instance level, the key points b1, b2 in **b** are similar to a3, a4 in **a**, and c2, c3 in **c** are similar to a1, a2 in **a**. In the function D_{min}, the distances between pictures are $D_{min}(a,b)=d(a3,b1)$ and $D_{min}(a,c)=d(a1,c2)$ (The black lines in Fig. 2(b) are pairwise distances calculated by D_{min}), and both are relative small. We will make the wrong judge if the latter is slightly smaller than the former one. So similar instances from different classes can weaken the quality of D_{min}. In the function D_{ave}, all the distances of pairwise instances will be computed and averaged, bringing negative information, such as $d(a3,b3), d(a4,b4), d(a3,b4)$, which will not be calculated in D_{am} (In Fig. 2(b), the red lines are redundant distance information in computing bag distance).

Algorithm 1 Single view multi-instance metric learning (SVMIML)

```
Input: Dataset $T = \{(X_i, y_i)\}_{i=1}^m$, $X_i = \{x_{i1}, x_{i2}, \cdots, x_{im_i}\}$, $y_i \in \{1,2,\cdots,c\}$, penalty parameters $\lambda$, learning rate $\eta$, maximum of iterations $R$.
Output: Target metrics $M^*$;
1: Initialize: $M^{(0)}$ as identity matrix and set $r=0$;
2: while not converge do
3:     Randomly choose $p$ pairs of samples: $\{(X_i, y_i)\}_{i=1}^p$ and build model (15);
4:     Calculate $\frac{\partial E_s}{\partial M^{(r)}}$ according to (15);
5:     Update model parameters $M^{(r+1)}$ using
           $$M^{(r+1)} = M^{(r)} - \eta \frac{\partial E_s}{\partial M^{(r)}}; \quad (17)$$
6:     $r = r+1$;
7: end while
8: Project $M^{(R)}$ by
       $$M^{(R)} = \mathrm{PSD}(M^{(R)}), \quad (18)$$
   where PSD denotes the projection operator of positive semi-definite space;
9: Return: Optimal $M^*$.
```

In multi-view case, problem (14) involve complex optimization with respect to $\{M_k\}_{k=1}^v$ and $\{\alpha_k\}_{k=1}^v$. Similar to Tang et al. (2021b), we can decompose (14) into two sub-problems respectively in an alternating optimization procedure. Iterations are repeated until convergence or a maximum number of iterations is reached.

Updating $\{M_k\}_{k=1}^v$. For the fixed $\alpha_k, k=1,\dots,v$, the gradient of $E(\mathscr{M},\alpha)$ with respect to M_k is computed as

$$\frac{\partial E}{\partial M_k} = \frac{1}{m}\sum_{i=1}^m \frac{\partial f(X_i,y_i)}{\partial M_k} + \lambda M_k + \frac{1}{m}\sum_{i=1}^m \frac{\sum_{y=1}^c \exp(-f(X_i,y))\frac{\partial f(X_i,y)}{\partial M_k}}{\sum_{y=1}^c \exp(-f(X_i,y))}. \quad (19)$$

To obtain $\frac{\partial f(X_i,y)}{\partial M_k}$, we decompose it as

$$\begin{aligned}\frac{\partial f(X_i,y)}{\partial M_k} &= \frac{\partial}{\partial M_k}\alpha_k^* D_{am}(X_i^k, X_{j^*}^k)\\ &= \frac{\alpha_k^*}{m_i^k}\sum_{p=1}^{m_i^k}\frac{\partial}{\partial M_k}\min_{x_{jl}\in X_j} d_M(x_{ip},x_{jl}) + \frac{\alpha_k^*}{m_j^k}\sum_{q=1}^{m_j^k}\frac{\partial}{\partial M_k}\min_{x_{ih}\in X_i} d_M(x_{jq},x_{ih})\\ &= \frac{\alpha_k^*}{m_i^k}\sum_{p=1}^{m_i^k}(x_{ip}-x_{jl^*})(x_{ip}-x_{jl^*})^\top + \frac{\alpha_k^*}{m_j^k}\sum_{q=1}^{m_j^k}(x_{jq}-x_{ih^*})(x_{jq}-x_{ih^*})^\top,\end{aligned} \quad (20)$$

where

$$(\alpha_k^*, j^*) = \arg\min_{\alpha,j}\sum_{k=1}^v \alpha_k D_{am}(X_i^k, X_j^k). \quad (21)$$

For each $p=1,\dots,m_i^k$,

$$l^* = \arg\min_{x_{jl}\in X_j} d_M(x_{ip},x_{jl}), \quad (22)$$

and for each $q=1,\dots,m_j^k$,

$$h^* = \arg\min_{x_{ih}\in X_i} d_M(x_{jq},x_{ih}). \quad (23)$$

Updating α. For the fixed $M_k,\ k=1,\dots,v$, the gradient of $E(\mathscr{M},\alpha)$ with respect to α is computed as

$$\frac{\partial E}{\partial \alpha} = \frac{1}{m}\sum_{i=1}^m F(X_i,y_i) + \mu\alpha + \frac{1}{m}\sum_{i=1}^m \frac{\sum_{y=1}^c \exp(-f(X_i,y))F(X_i,y)}{\sum_{y=1}^c \exp(-f(X_i,y))}, \quad (24)$$

where

$$F(X_i,y) = (D_{am}(X_i^1, X_j^1; M_1),\dots,D_{am}(X_i^v, X_j^v; M_v))^\top, \quad (25)$$

and

$$(X_j^1,\dots,X_j^v) = \arg\min_{y_j=y}\sum_{k=1}^v \alpha_k D_{am}(X_i^k, X_j^k; M_k). \quad (26)$$

The detailed procedure of our method (14) is shown in Algorithm 2. Given an unknown image L with v views $X_l^1,\dots,X_l^v$, its label can be predicted by

$$y_l = \arg\max_{y=1,\dots,c} p(y|X_l^1,\dots,X_l^v;\mathscr{M}^*), \quad (27)$$

where $\mathscr{M}^* = \{M_1^*,\dots,M_v^*\}$.

3.5. Computational complexity

Our model (14) is solved iteratively with alternate optimization. The computation of mini-batch stochastic gradient descent is the main part of the computational cost. It contains two parts: updating metrics and weights. In the process of updating the metrics, the computational cost in every iteration is $O(V^2K^2mn(m+n))$, where V is the number of views, K is the number of images, m is the average number of instances in each bag and n is the average length of each instance. And the cost is $O(VK^2mn(m+n))$ in updating weights. Considering the iteration number R and Q, the total computational cost of our model is $O(RQV^2K^2mn(m+n))$. Due to the merits of mini-batch stochastic gradient descent, the number of iterations is reduced and the stable solution can be achieved more effective than the original stochastic gradient descent algorithm. Although the cost is relatively high, it is tractable in our experiments. It can further be improved by GPU acceleration or some specific speedup strategies such as parallel computing in the future.

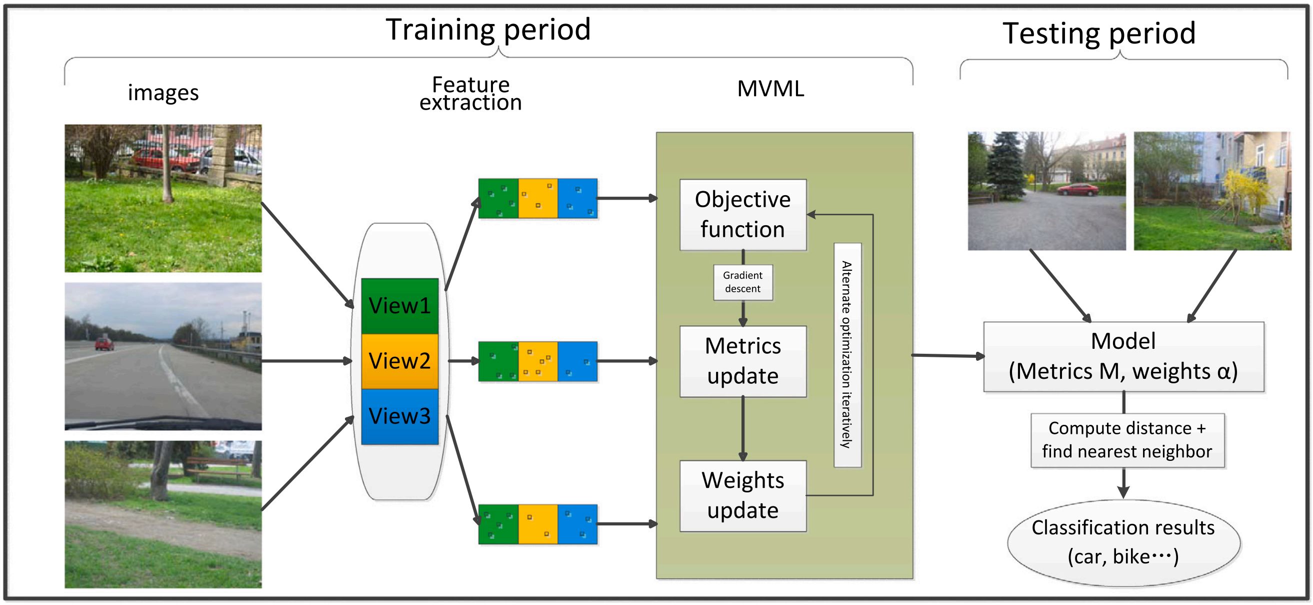

Training period
Testing period
images
Feature extraction
MVML
View1
View2
View3
Objective function
Gradient descent
Metrics update
Weights update
Alternate optimization iteratively
Model (Metrics M, weights α)
Compute distance + find nearest neighbor
Classification results (car, bike⋯)

Fig. 3. The workflow of our MVMIML method. The model can be divided into two periods: training period and testing period.

Algorithm 2 Multi-view multi-instance metric learning (MVMIML)

```
Input: Dataset $T^k = \{(X_i^k, y_i)\}_{i=1}^m$, $X_i^k = \{x_{i1}^k, x_{i2}^k, \cdots, x_{im_i^k}^k\}$, $y_i \in \{1, 2, \cdots, c\}$, $k = 1, 2, \cdots, v$, penalty parameters $\lambda$, learning rate $\eta_1$, $\eta_2$, maximum of iterations $R$, $Q$.
Output: Target metrics $M_1^*, \cdots, M_v^*$ and weights $\alpha^* = (\alpha_1^*, \cdots, \alpha_v^*)^\top$;
Initialize: $M_1^{(0)}, M_2^{(0)}, \cdots, M_v^{(0)}$ as identity matrix, $\alpha^{(0)} = (\alpha_1^{(0)}, \alpha_2^{(0)}, \cdots, \alpha_v^{(0)})^\top = (\frac{1}{v}, \cdots, \frac{1}{v})^\top$ and set $r = 0$, $q = 0$;
while not converge do
    while not converge do
        Randomly choose $p$ pairs of samples: $\{(X_i^k, y_i)\}_{i=1}^p$ $(k = 1, 2, \cdots, v)$ and build model (14);
        Calculate $\frac{\partial E}{\partial M_k^{(r)}}$ $(k = 1, 2, \cdots, v)$ by using (19) and (20);
        Update model parameters $M_k^{(r+1)}$ $(k = 1, 2, \cdots, v)$ by using
            $$M_k^{(r+1)} = M_k^{(r)} - \eta_1 \frac{\partial E}{\partial M_k^{(r)}};\qquad (28)$$
        $r = r + 1$;
    end while
    Project $M_k^{(R)}$ by
        $$M_k^{(R)} = \mathrm{PSD}(M_k^{(R)}),\qquad (29)$$
    where PSD denotes the projection operator of positive semi-definite space, and $k = 1, 2, \cdots, v$;
    Calculate $\frac{\partial E}{\partial \alpha^{(q)}}$ by using (24) with fixed $M_1^{(R)}, M_2^{(R)}, \cdots, M_v^{(R)}$;
    Update $\alpha^{(q+1)}$ by using
        $$\alpha^{(q+1)} = \alpha^{(q)} - \eta_2 \frac{\partial E}{\partial \alpha^{(q)}};\qquad (30)$$
    $q = q + 1$;
end while
Return: Optimal target metrics $M_1^*, \cdots, M_v^*$ and weights $\alpha^* = (\alpha_1^*, \cdots, \alpha_v^*)^\top$.
```

4. Experiments

This section presents experimental results on both the image datasets to verify the effectiveness of MVMIML. The experiments are performed on Matlab 2015a (PC, 8 GB RAM). Besides, for the deep learning experiment section, we use the TensorFlow framework (GPU: NVIDIA M40).

4.1. Datasets

Six datasets *Corel, Caltech, Birds, Butterfly, Galaxy Zoo* and *FERET* are selected and each dataset is converted into standard format containing three parts, i.e., features, bag ids and labels. These datasets can be divided into four categories: (1) Object detection. Detect particular object in an image by the unique features of the object. The object often appears with complex background that may affect the feature extraction. (2) Species recognition. A species may contain several classes. The task is to recognize the class of a species. The difficulty is that there exists exiguous difference between distinct classes of the same species. (3) Galaxy discrimination. Discriminating the shape of galaxy automatically is useful in astronomy since it can help scientists track the evolution of galaxies. (4) Face identification. *FERET* (Facial Recognition Technology) dataset (Phillips et al., 2000) is used to identify the gender of a given face. Its primary task is to develop automatic face recognition technology to assist security, intelligence and law enforcement personnel. For these four categories, some example are presented as shown in Figs. 4, 5, 6 and 7. The detailed descriptions of these datasets are provided as shown in Table 1.

4.2. Feature extraction for images

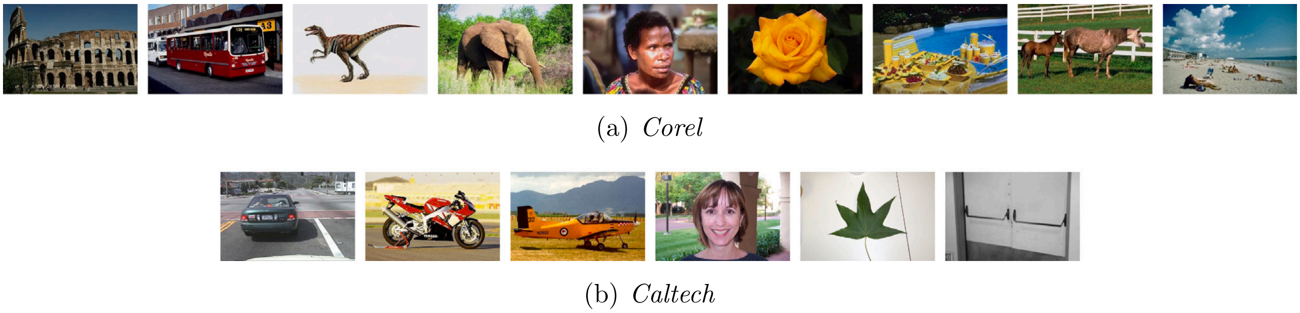
In this section, we introduce three features extraction methods to construct different views of the image datasets.

4.2.1. HOG feature

It first divides image into smaller “cells” and accumulates histogram of gradient or edge directions for each cell (Dalal & Triggs, 2005). The combination of these histograms is the HOG feature of the whole image. However, we describe each image with a bag of 9-size cells instead of combining the histograms. Each cell is represented by a feature vector.

Table 1
Characteristics of image datasets.

Dataset	Task	# of Classes	Class names
Corel (Duygulu et al., 2002)	Object detection	10	architecture, bus, dinosaur, elephant, face, flower, food, horse, sky and snowberg
Caltech (Bosch et al., 2007)	Object detection	6	car, motorcycle, airplane, face, leaf and background
Butterfly (Lazebnik et al., 2004)	Species recognition	7	admiral, black-swallowtail, machaon, monarch-closed, monarch-open, peacock and zebra
Birds (Mohanty et al., 2020)	Species recognition	6	egret, mandarin, owl, puffin, toucan and wood duck
Galaxy Zoo (Misra et al., 2020)	Galaxy discrimination	3	edge-on, elliptical and spiral
FERET (Phillips et al., 2000)	Face identification	2	man and woman


(a) *Corel*

(b) *Caltech*

Fig. 4. Images of Object Detection.

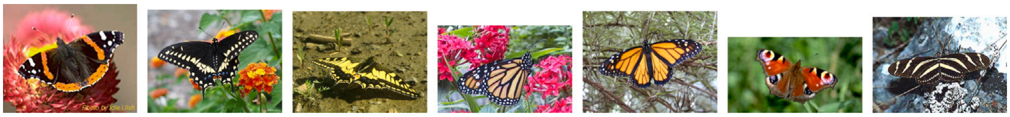


Fig. 5. Images of the *Butterfly*.

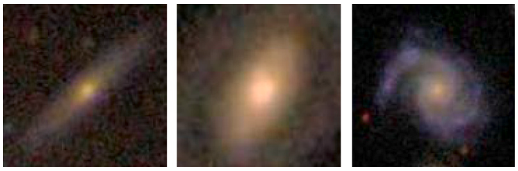


Fig. 6. Images of the *Galaxy Zoo*.

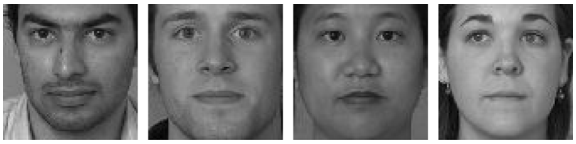


Fig. 7. Images of the *FERET*.

4.2.2. SIFT key points

The SIFT features are suitable for multi-instance learning. SIFT has been widely applied to many applications (Felzenszwalb et al., 2010) since it is invariant to image scale and rotation. SIFT (Lowe, 2004) can find interest points at multiple scale to represent important regions of each image. Each key point is a 128-length numerical feature vector and each image is described by a bag of multiple key points. It is worthy to point out that SIFT can extract different numbers of key points from diverse images, so that the idea of concatenating the vectors into a single feature vector is not tractable in traditional classification.

4.2.3. Uniform patches with LBP

Similar to "visual dictionary" (Wen et al., 2009), every image will be divided into 4×4 uniformly distributed patches, since multi-instance feature can be more powerful than a single instance in representing an image (González et al., 2017). Every patch can be expressed by a single instance with the 256-dimensional LBP features. Then each image is transformed into a bag including 16 instances.

Table 2
The computed distances in three kinds of features.

	D_{am}			D_{min}			D_{ave}		
	0	$\underline{0.49}$	$\underline{0.41}$	0	$\underline{0.29}$	$\underline{0.26}$	0.38	$\underline{0.81}$	$\underline{0.61}$
HOG	$\underline{0.49}$	0		$\underline{0.29}$	0	1.03	$\underline{0.81}$	0.37	1.3
	$\underline{0.41}$	1.14	0	$\underline{0.26}$	1.03	0	$\underline{0.61}$	1.3	0.34
	0	**0.66**	0.78	0	**0.30**	0.48	0.94	$\underline{1.08}$	$\underline{1.07}$
SIFT	**0.66**	0	0.72	**0.30**	0	0.45	$\underline{1.08}$	0.95	$\underline{1.07}$
	0.78	0.72	0	0.48	0.45	0	$\underline{1.07}$	$\underline{1.07}$	0.94
	0	**1.69**	1.96	0	$\underline{0.49}$	$\underline{0.38}$	3.86	$\underline{4.41}$	4.57
LBP	**1.69**	0	2.21	$\underline{0.49}$	0	0.67	$\underline{4.41}$	2.82	$\underline{4.37}$
	1.96	2.21	0	$\underline{0.38}$	0.67	0	4.57	$\underline{4.37}$	3.07

4.3. Distance comparison

In Section 3.2, we have claimed that the distance function D_{am}, designed for bags, is prior to previous D_{ave} and D_{min}. Next, we will perform numerical experiments to verify the judgment. First, a toy example is given in Fig. 8 to show the difference among three distance functions. We select three images from the *Corel* dataset. The former two images I_1 and I_2 belong to the class of *architecture* and the third one I_3 belongs to *snowberg*. Then, the features of HOG, SIFT and LBP are extracted and depicted in the last three lines of Fig. 8. Each curve in the subfigures represents an instance. The horizontal and vertical axes denote the number of the features and the value of the components of the instance, respectively. For the subfigures about the first two images, the trends of the curves are similar for the HOG, SIFT and LBP features, which verify that the first two images belong to the same class. Further more, we compute the distances between any two images in Table 2. Since the first two images belong to the same class, the distances that verify the fact are in boldface. The numbers with underline can

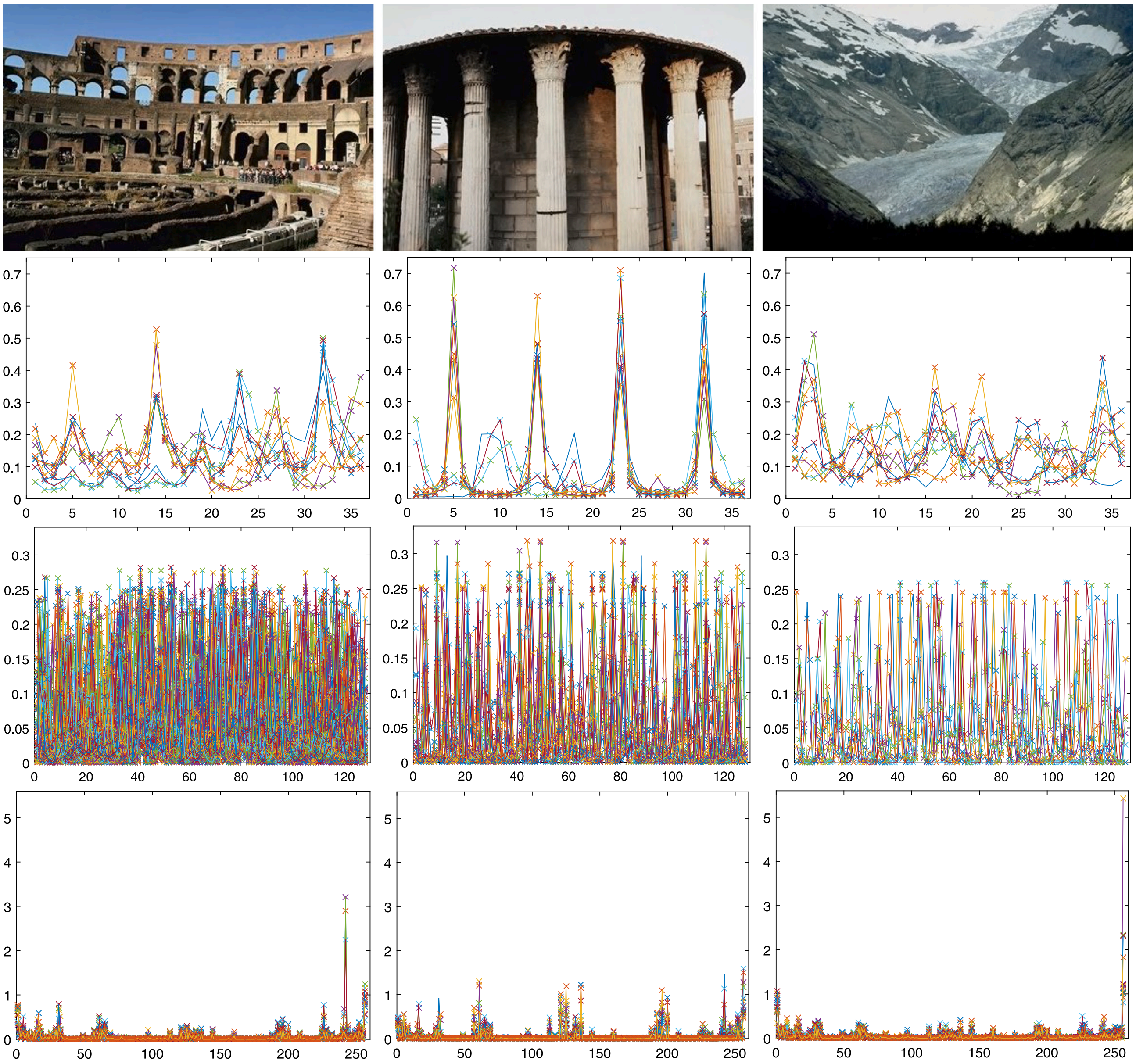


Fig. 8. A toy example to compare three distance functions D_{am}, D_{min} and D_{ave}. The second, third and fourth line of images correspond to HOG, SIFT, and LBP features respectively. Each curve in a subfigure denotes an instance. The horizontal axis denotes the order number of each feature and the vertical axis denotes the numerical value of the components of the instance.

Table 3
1-NN classification accuracy of different distances on different features.

Datasets	Distance	Feature			Average
		HOG	SIFT	LBP	
Car (600&2)	D_{ave}	56.67 ± 2.25	51.83 ± 1.04	47.50 ± 6.56	52.00
	D_{min}	60.67 ± 2.52	59.33 ± 2.36	53.83 ± 1.61	57.94
	D_{am}	**64.00 ± 3.46**	**64.50 ± 2.00**	**59.17 ± 1.89**	**62.56**
Butterfly (619&7)	D_{ave}	17.28 ± 3.64	23.42 ± 3.16	21.16 ± 1.97	20.62
	D_{min}	40.55 ± 2.75	**84.33 ± 0.98**	26.66 ± 1.31	50.51
	D_{am}	**52.66 ± 1.89**	81.74 ± 1.49	**28.12 ± 4.73**	**54.17**
Corel (1000&10)	D_{ave}	15.20 ± 4.53	13.60 ± 0.44	36.50 ± 3.45	21.77
	D_{min}	44.50 ± 1.28	32.90 ± 1.08	39.80 ± 3.05	39.07
	D_{am}	**50.70 ± 3.47**	**46.30 ± 3.06**	**58.00 ± 1.32**	**51.67**

misguide the judgment of distance function. Indeed, these three images are similar in structure, color and luminance, which gives a challenge to the distance function. In SIFT and LBP features, $D_{am}(I_1, I_2)$ is smaller than $D_{am}(I_1, I_3)$ and $D_{am}(I_2, I_3)$, implying that I_1 and I_2 belong to the same class. So D_{am} can make right judges in SIFT and LBP features, better than D_{ave} and D_{min}.

To further validate the effectiveness of D_{am}, 1NN classification is implemented with these three distance functions. Euclidean distance

Table 4
Classification accuracy of multi-instance methods and MVMIML.

Datasets	Single view				Multi-view(MVMIML)			
	Method	HOG	SIFT	LBP	H&S	H&L	S&L	H&S&L
Corel (300&10)	C-kNN(min)	33.00 ± 6.08	26.33 ± 4.04	33.33 ± 4.51	52.33 ± 6.11	55.33 ± 2.52	57.67 ± 6.43	**60.00** ± **1.73**
	C-kNN(max)	29.67 ± 7.51	22.00 ± 4.36	43.67 ± 5.13				
	MInD	35.33 ± 3.79	25.67 ± 4.04	23.33 ± 1.53				
	kNN + D_{am}	43.00 ± 2.65	42.67 ± 4.16	54.00 ± 2.00				
	SVMIML	44.00 ± 1.00	42.98 ± 3.32	59.33 ± 5.77				
Caltech (300&6)	C-kNN(min)	64.33 ± 6.81	55.67 ± 14.5	54.00 ± 8.72	86.00 ± 2.65	85.00 ± 6.08	77.33 ± 1.53	**87.00** ± **4.58**
	C-kNN(max)	37.67 ± 13.3	46.67 ± 4.93	59.00 ± 2.00				
	MInD	77.67 ± 1.15	55.33 ± 3.79	65.67 ± 4.73				
	kNN + D_{am}	79.98 ± 2.77	66.59 ± 1.79	78.67 ± 3.21				
	SVMIML	80.33 ± 2.52	67.67 ± 1.53	78.33 ± 3.06				
Birds (600&6)	C-kNN(min)	23.00 ± 3.04	35.00 ± 4.27	19.33 ± 1.61	42.50 ± 3.28	33.67 ± 3.06	44.50 ± 2.29	43.17 ± 2.52
	C-kNN(max)	20.67 ± 2.36	23.33 ± 2.08	26.50 ± 1.73				
	MInD	28.33 ± 2.25	35.17 ± 4.01	27.17 ± 5.06				
	kNN + D_{am}	34.17 ± 2.02	44.32 ± 2.93	32.99 ± 4.25				
	SVMIML	32.67 ± 3.62	**44.67** ± **3.06**	33.67 ± 3.01				
Butterfly (280&7)	C-kNN(min)	37.49 ± 3.01	58.90 ± 12.1	21.79 ± 1.74	73.22 ± 0.93	43.25 ± 8.13	67.14 ± 4.15	**73.57** ± **2.29**
	C-kNN(max)	33.58 ± 2.42	32.86 ± 1.35	19.99 ± 6.25				
	MInD	44.29 ± 5.58	60.33 ± 5.89	16.44 ± 3.51				
	kNN + D_{am}	47.51 ± 2.93	71.43 ± 1.34	22.85 ± 1.50				
	SVMIML	48.22 ± 2.02	71.79 ± 2.44	24.64 ± 2.15				
Galaxy (210&3)	C-kNN(min)	57.62 ± 2.18	49.52 ± 3.60	57.62 ± 7.05	81.90 ± 4.12	83.33 ± 3.60	84.29 ± 5.71	**85.71** ± **3.78**
	C-kNN(max)	62.38 ± 2.18	51.43 ± 9.37	74.76 ± 1.65				
	MInD	77.62 ± 0.82	65.24 ± 0.82	71.43 ± 6.55				
	kNN + D_{am}	77.92 ± 1.35	62.38 ± 1.65	81.90 ± 8.37				
	SVMIML	78.10 ± 1.82	62.86 ± 1.43	82.38 ± 5.87				
FERET (150&2)	C-kNN(min)	67.33 ± 4.62	62.67 ± 3.06	63.33 ± 8.08	84.00 ± 8.72	81.33 ± 1.15	78.00 ± 4.00	**84.00** ± **2.00**
	C-kNN(max)	76.00 ± 2.00	62.67 ± 5.03	59.33 ± 6.43				
	MInD	82.00 ± 2.00	73.33 ± 6.43	73.33 ± 1.15				
	kNN + D_{am}	82.55 ± 2.57	76.00 ± 7.21	71.33 ± 5.03				
	SVMIML	82.67 ± 2.31	76.00 ± 5.29	76.00 ± 2.00				

is used to compute the distance between instances. We select three datasets, *Car* (from Caltech), *butterfly* and *Corel*, and apply three-fold cross validation in this experiments. Accuracy and standard deviation are reported in Table 3. It is obvious that 1NN with D_{am} achieves the best performance on most of the features of the three datasets. D_{ave} performs much worse than D_{am} and D_{min}.

4.4. Performance evaluation

In this section, we will evaluate our model from three aspects, including classification ability in different scenarios, robustness to parameters and sensitivity to instance number of bags.

4.4.1. Image classification

To evaluate the performance of MVMIML comprehensively, three categories of methods will be selected to make comparisons, i.e., multi-instance learning, metric learning and multi-instance metric learning. For each method, we conduct experiments to yield the average accuracy on six datasets, and all the experiments are repeated 10 times.

A. Compared with multi-instance learning

The experiments are divided into two parts: single view and multi-view. In single view classification, experiments on three features HOG, SIFT and LBP are conducted independently. Three baseline methods are selected. For the Citation kNN method (Wang & Zucker, 2000), we set $R = 3$ and $C = 5$, and apply the minimal and maximal Hausdorff distance respectively. Note that Citation-kNN is simplified as C-kNN in the following. The second baseline method is MInD (Cheplygina et al., 2015) associated with asymmetric average minimum distances. The third one is the kNN classification with D_{am} distance measure. In SVMIML, the penalty parameter λ and the learning rate η are both empirically set to be 0.1 and the number of iteration R is set to be 3. From Table 4, SVMIML always behaves the best performance.

For multi-view classification, the three features can be combined into four groups: HOG + SIFT (H&S), HOG + LBP (H&L), SIFT + LBP (S&L) and HOG + SIFT + LBP (H&S&L). The experiments on these four multi-view cases are performed individually. To obtain the best parameters for all the methods, the grid search strategy with the five-fold cross-validation technique is implemented. The penalty parameters λ and μ are both selected from the set $\{0.01, 0.1, 1\}$ and the combination of η_1 and η_2 is chosen from the set $\{(0.01, 0.01), (0.05, 0.05), (0.1, 0.1)\}$. The numbers of iteration τ and R are set to be 2 and 3 respectively. The classification results are shown in Table 4. The experiments on the feature group H&S&L behave the best on 5 out of 6 image datasets, better than single view classification, despite metric learning or not. Although MVMIML in the *Birds* dataset is not top-ranked, they are close to the best results. Therefore, MVMIML improves the classification performance of multi-instance learning. Further, six images from three datasets are displayed in Table 5 with their corresponding nearest images under different views. It implies that our method can truly find a data-dependent metric to make similar images closer and boost the classification performance.

B. Compared with metric-learning

Traditionally, metric learning methods are proposed for standard classification, in which each pattern is a single feature vector. However, in the above feature extraction, each image is represented by a bag with multiple instances. To adapt the metric learning scenario, we apply the framework of bag-of-words (Li & Perona, 2005) to transform each image into a single vector. Five classical metric learning methods, i.e., ITML (Davis et al., 2007), LMNN (Weinberger et al., 2005), BoostMetric (Shen et al., 2009), distance metric learning with eigenvalue optimization (DML-eig) (Ying & Li, 2012) and SERAPH (Niu et al., 2014) are selected and kNN with Euclidean distance (Eucl) is used as a baseline. The classification performances of these methods are shown in Table 6. The best results of single-view methods and of all the methods are highlighted in italics and in bold respectively. According to Table 6,

Table 5
Performance of the images associated with its nearest images in different views.

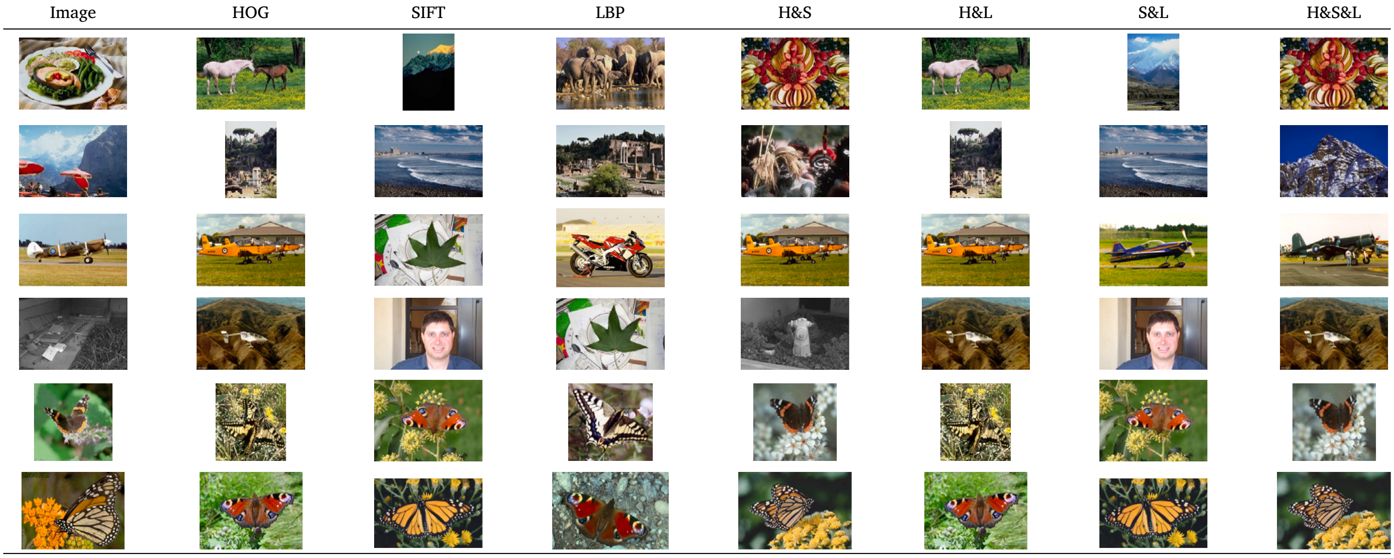


Table 6
Classification accuracy of metric learning methods on single-view and multi-view.

Datasets	View	Eucl	ITML	LMNN	BoostMetric	DML-eig	SERAPH	SVMIML	MVMIML
Corel (300&10)	HOG	42.33 ± 4.73	48.33 ± 1.53	46.00 ± 3.46	40.00 ± 2.00	34.00 ± 3.61	*47.33 ± 3.21*	44.00 ± 1.00	**60.00 ± 1.73**
	SIFT	21.67 ± 2.89	28.33 ± 2.89	27.00 ± 3.61	28.67 ± 4.51	18.33 ± 2.08	26.33 ± 2.89	*42.98 ± 3.32*	
	LBP	51.33 ± 5.13	57.67 ± 9.29	59.67 ± 4.16	46.67 ± 4.16	38.00 ± 4.36	58.00 ± 7.55	*59.33 ± 5.77*	
Caltech (300&6)	HOG	80.33 ± 1.53	82.67 ± 0.58	83.33 ± 0.58	80.67 ± 2.08	66.67 ± 1.53	*86.33 ± 2.08*	80.33 ± 2.52	**87.00 ± 4.58**
	SIFT	42.67 ± 3.21	62.33 ± 6.51	67.00 ± 4.58	61.00 ± 6.08	47.67 ± 8.96	66.33 ± 5.51	*67.67 ± 1.53*	
	LBP	74.67 ± 4.16	79.00 ± 3.00	80.67 ± 2.08	77.33 ± 8.33	57.00 ± 2.00	*80.67 ± 2.52*	78.33 ± 3.06	
Birds (600&6)	HOG	26.33 ± 2.93	30.67 ± 6.58	28.00 ± 1.80	25.83 ± 2.25	23.33 ± 2.52	30.17 ± 2.02	*32.67 ± 3.62*	43.17 ± 2.52
	SIFT	33.00 ± 8.89	32.17 ± 3.75	40.00 ± 8.05	35.50 ± 2.65	27.00 ± 5.07	40.00 ± 5.77	**44.67 ± 3.06**	
	LBP	27.50 ± 0.50	28.50 ± 3.12	30.17 ± 0.57	25.17 ± 2.57	22.83 ± 4.37	29.83 ± 2.57	*33.67 ± 3.01*	
Butterfly (280&7)	HOG	46.42 ± 4.82	48.21 ± 2.65	42.16 ± 7.29	38.56 ± 2.69	29.27 ± 7.66	43.58 ± 7.01	*48.22 ± 2.02*	**73.57 ± 2.29**
	SIFT	49.99 ± 5.12	56.42 ± 1.39	54.98 ± 4.12	57.85 ± 1.71	35.00 ± 6.53	55.70 ± 2.92	*71.79 ± 2.44*	
	LBP	20.01 ± 1.75	20.37 ± 2.25	26.06 ± 3.96	17.16 ± 4.99	20.72 ± 2.72	*25.35 ± 5.24*	24.64 ± 2.15	
Galaxy (210&3)	HOG	75.24 ± 5.77	75.24 ± 2.97	75.24 ± 4.59	78.10 ± 2.97	60.48 ± 3.60	76.67 ± 3.60	*78.10 ± 1.82*	**85.71 ± 3.78**
	SIFT	55.24 ± 6.75	57.14 ± 5.15	67.62 ± 0.82	67.14 ± 2.86	47.62 ± 1.65	*68.57 ± 1.43*	62.86 ± 1.43	
	LBP	63.33 ± 1.65	64.76 ± 3.60	69.05 ± 8.12	66.19 ± 2.18	59.52 ± 4.59	71.43 ± 4.95	*82.38 ± 5.87*	
FERET (150&2)	HOG	70.67 ± 9.02	78.00 ± 4.00	78.67 ± 4.16	78.00 ± 2.00	62.00 ± 8.72	80.00 ± 5.29	*82.67 ± 2.31*	**84.00 ± 2.00**
	SIFT	67.33 ± 2.31	66.00 ± 2.00	71.33 ± 7.57	64.00 ± 6.93	62.00 ± 4.00	74.67 ± 10.3	*76.00 ± 5.29*	
	LBP	64.00 ± 6.93	54.67 ± 13.3	71.33 ± 4.62	70.00 ± 4.93	62.67 ± 7.57	72.00 ± 5.29	*76.00 ± 2.00*	

our proposed SVMIML model behaves the best on 13 out of 18 (72.22%) cases for all the datasets with diverse features. To incorporate three views, MVMIML obtain the highest results on 17 out of 18 the cases (94.44%).

C. Compared with multi-instance metric learning

We compare the performance of MVMIML with that of two multi-instance metric learning methods in this section. The first one is MIMEL which introduces the idea of ITML (Davis et al., 2007) into multi-instance problem (Xu et al., 2011). And the second one is metric learning for multi-instance with collapsed bags (MIMLCB), where k−means is employed to get collapsed bags and the idea of maximizing relative bag distance is applied on the transformed bags (Li et al., 2017). The classification results are recorded in Table 7. From this table, we can observe that MVMIML yields better performance than MIMEL and MIMLCB. The reason may be that the information from multiple views can contribute to the performance improvement. MIMEL behaves much worse than MIMLCB, which may result from the application of distance function D_{ave}.

4.4.2. Ablation analysis

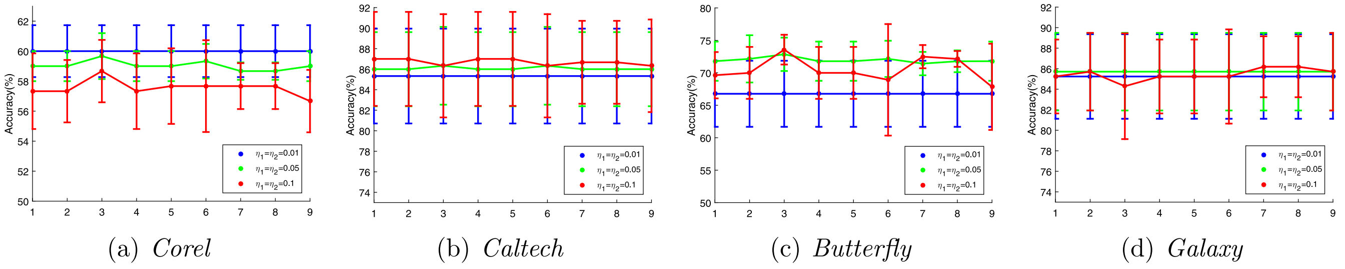
To clearly explain the individual contribution of metric learning and multi-view learning, ablation analysis will be made based on the results in Table 4. In single-view learning, SVMIML performs better than $k\mathrm{NN} + D_{am}$ in most cases, which verifies the effectiveness of metric learning in single view multi-instance learning. Metric learning can be used to build bridges between instances, and the designed distance measure D_{am} is efficient. Comparing SVMIML and MVMIML, the technique of metric learning is both implemented, but different in the number of views. It can be seen that MVMIML with multiple views often achieves higher accuracy than SVMIML. It indicates that our method can extract useful information from all the views and assemble them effectually, based on the inference that different features are complementary to each other.

4.4.3. Influence of parameters

In this section, the influence of parameters λ and μ and learning rates η_1 and η_2 are explored. The iteration numbers Q and R are the same as the above experiments. As previously mentioned, λ and μ are both selected from the set $\{0.01, 0.1, 1\}$. Fixed the combination of

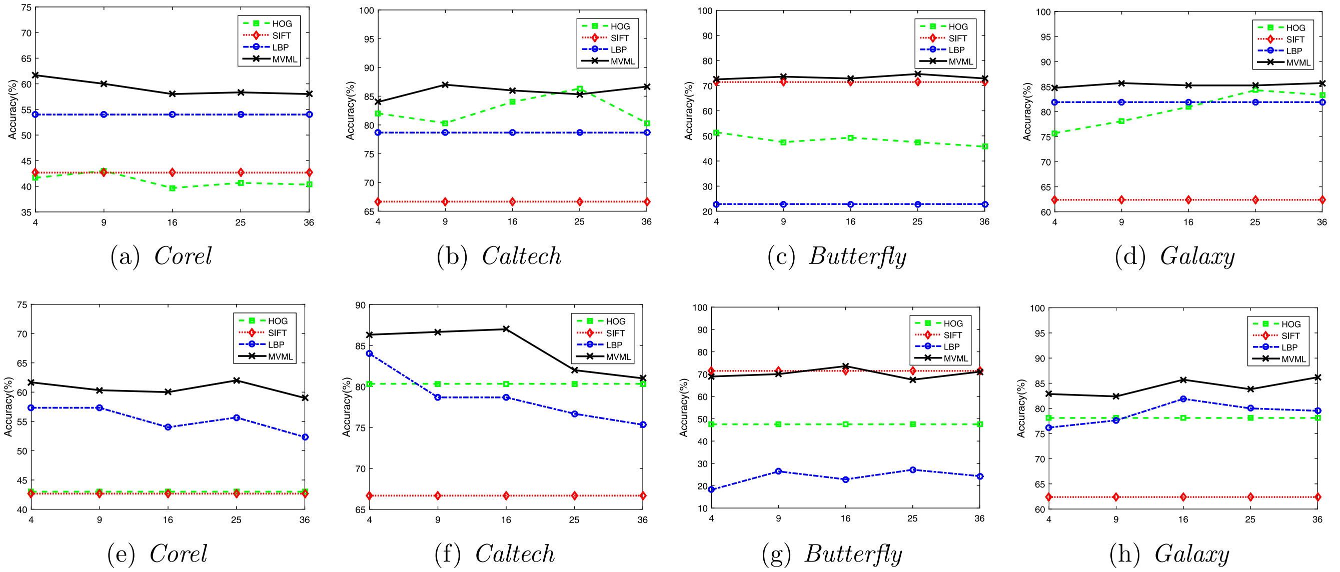
Table 7
Comparison of the classification accuracy of multi-instance metric learning methods and MVMIML.

Datasets	Single view				Multi-view (MVMIML)			
	Method	HOG	SIFT	LBP	H&S	H&L	S&L	H&S&L
Corel (300&10)	MIMEL MIMLCB	25.00 ± 7.81 43.33 ± 3.06	21.33 ± 5.69 43.00 ± 4.58	17.00 ± 9.17 54.00 ± 2.00	52.33 ± 6.11	55.33 ± 2.52	57.67 ± 6.43	**60.00 ± 1.73**
Caltech (300&6)	MIMEL MIMLCB	46.67 ± 4.93 80.00 ± 3.00	22.00 ± 5.00 66.67 ± 1.53	35.67 ± 5.69 78.67 ± 3.21	86.00 ± 2.65	85.00 ± 6.08	77.33 ± 1.53	**87.00 ± 4.58**
Birds (600&6)	MIMEL MIMLCB	18.00 ± 3.04 34.33 ± 2.25	27.33 ± 3.25 44.50 ± 3.12	17.00 ± 1.00 33.83 ± 3.06	42.50 ± 3.28	33.67 ± 3.06	**44.50 ± 2.29**	43.17 ± 2.52
Butterfly (280&7)	MIMEL MIMLCB	12.85 ± 2.82 47.86 ± 2.44	28.94 ± 4.76 70.71 ± 1.78	18.22 ± 1.92 22.85 ± 1.50	73.22 ± 0.93	43.25 ± 8.13	67.14 ± 4.15	**73.57 ± 2.29**
Galaxy (210&3)	MIMEL MIMLCB	49.05 ± 8.12 78.10 ± 0.82	44.29 ± 3.78 62.38 ± 1.65	43.33 ± 13.5 81.90 ± 8.37	81.90 ± 4.12	83.33 ± 3.60	84.29 ± 5.71	**85.71 ± 3.78**
FERET (150&2)	MIMEL MIMLCB	50.00 ± 3.46 82.67 ± 2.31	56.67 ± 17.2 76.00 ± 7.21	50.00 ± 3.46 71.33 ± 5.03	84.00 ± 8.72	81.33 ± 1.15	78.00 ± 4.00	**84.00 ± 2.00**



(a) *Corel* (b) *Caltech* (c) *Butterfly* (d) *Galaxy*

Fig. 9. Influence of the penalty parameters and learning rates.



(a) *Corel* (b) *Caltech* (c) *Butterfly* (d) *Galaxy*

(e) *Corel* (f) *Caltech* (g) *Butterfly* (h) *Galaxy*

Fig. 10. Influence of instance number in HOG and LBP features. In all the subfigures, the instance numbers of SIFT feature are the same as previous classification. In (a), (b), (c) and (d), the instance numbers of LBP feature are all fixed as 16. In (e), (f), (g) and (h), the instance numbers of HOG feature are all fixed as 9.

(η_1, η_2) as $(0.01, 0.01)$, $(0.05, 0.05)$ and $(0.1, 0.1)$, the accuracy curve and corresponding standard deviation with respect to the combination of λ and μ are plotted in Fig. 9. Each subfigure corresponds to a selected dataset. The axis of x denotes nine combinations of λ and μ. For every combination of η_1 and η_2, the accuracy fluctuates slightly, which indicates that our model is robust to the penalty parameters when they are sufficiently small. For each combination of λ and μ, the accuracy does not change drastically with respect to η_1 and η_2. It implies that MVMIML is insensitive to the learning rates. The faint influence of parameters verifies the robustness of our proposed model.

4.4.4. Influence of instance number

In previous feature extraction, HOG and LBP are extracted as bag-of-words representation by dividing every image uniformly. In HOG and LBP feature, every image contains 9 and 16 instances respectively. Next, we will investigate the influence of instance number in each bag. The penalty parameters λ and μ are both set to be 0.01 and the learning rates η_1 and η_2 are both set to be 0.1. For HOG feature, each image is further divided into 4, 16, 25 and 36 cells and the accuracy curve of single view and multi-view (H&S&L) with respect to instance number is depicted in Fig. 10. We can observe that the accuracy of single view changes slightly and the accuracy of multi-view remains stable. Such observation verifies the robustness of our model to instance number of HOG feature. The model can extract information consistently from HOG feature, and immune to the instance number. For LBP feature, each image is further divided into 4, 9, 25 and 36 patches and the accuracy curve of single view and multi-view (H&S&L) with respect to instance number is reported in Fig. 10. The accuracy curve of multi-view (H&S&L) has similar fluctuation trend with the accuracy curve of single view, but both curves have small amplitudes. The experiments demonstrate that our model is not sensitive to instance number, resulting from that a bag with different instance numbers can be actually seem as the feature extraction in different scale, which will not affect metric learning.

5. Conclusions

In this paper, we propose a new multi-view multi-instance metric learning method named MVMIML for image classification, which integrates the merits of both the multi-view multi-instance representation and metric learning into a unified framework. Due to the merits of bag-of-words representations and multiple views, multi-instance multi-view features are extracted for each image. To combine multi-instance features effectively, we design a new distance function for bags, which calculates the weighted sum of the distance between bags from each single view. To guarantee that every image is similar to its nearest images, the joint conditional probability is maximized to pursue view-dependent metrics by using metric learning technique on multiple views. We then design the alternating iteration optimization algorithms to solve MVMIML, and analyze its computational complexity theoretically. The advantages of the newly-designed distance function and the effectiveness of MVMIML have been confirmed in the numerical experiments. This paper brings a new insight of utilizing the metric learning method to handle the image datasets with multi-view multi-instance representations. In future work, we will explore more kinds of features and design more efficient algorithm to solve our method. Extensions of MVMIML to the deep learning research field are also meaningful and need to be taken into account.

CRediT authorship contribution statement

Jingjing Tang: Methodology, Writing – Original Draft, Project administration, Theoretical analysis, Funding acquisition. **Dewei Li:** Software, Data curation, Writing – original draft. **Yingjie Tian:** Supervision, Writing – review & editing, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work has been partially supported by grants from National Natural Science Foundation of China (Nos. 71991472, 71901179, 12071458, 71731009 and 71910107002).

References

Ahonen, T., Hadid, A., & Pietikäinen, M. (2004). Face recognition with local binary patterns. In *Proceedings of the European conference on computer vision* (pp. 469–481). Springer.

Akaho, S. (2006). A kernel method for canonical correlation analysis. *International Meeting of the Psychometric Society*, *40*(2), 263–269.

Bay, H., Tuytelaars, T., & Van Gool, L. (2006). Surf: Speeded up robust features. In *Proceedings of the European conference on computer vision* (pp. 404–417). Springer.

Bosch, A., Zisserman, A., & Munoz, X. (2007). Image classification using random forests and ferns. In *Proceedings of the international conference on computer vision* (pp. 1–8). IEEE.

Cano, A. (2017). An ensemble approach to multi-view multi-instance learning. *Knowledge-Based Systems*, *136*, 46–57.

Cheplygina, V., Tax, D. M. J., & Loog, M. (2015). Multiple instance learning with bag dissimilarities. *Pattern Recognition*, *48*(1), 264–275.

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Proceedings of the international conference on computer vision and pattern recognition* (pp. 886–893). IEEE.

Davis, J. V., Kulis, B., Jain, P., Sra, S., & Dhillon, I. S. (2007). Information-theoretic metric learning. In *Proceedings of the international conference on machine learning* (pp. 209–216). ACM.

Dietterich, T. G., Lathrop, R. H., & Lozano-Pérez, T. (1997). Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, *89*(1), 31–71.

Duygulu, P., Barnard, K., de Freitas, J. F., & Forsyth, D. A. (2002). Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *Proceedings of the European conference on computer vision* (pp. 97–112). Springer.

Felzenszwalb, P. F., Girshick, R. B., McAllester, D., & Ramanan, D. (2010). Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *32*(9), 1627–1645.

Globerson, A., & Roweis, S. T. (2005). Metric learning by collapsing classes. In *Advances in neural information processing systems* (pp. 451–458).

Goldberger, J., Hinton, G. E., Roweis, S. T., & Salakhutdinov, R. (2004). Neighbourhood components analysis. In *Advances in neural information processing systems* (pp. 513–520).

González, L. C., Moreno, R., Escalante, H. J., Martínez, F., & Carlos, M. R. (2017). Learning roadway surface disruption patterns using the bag of words representation. *IEEE Transactions on Intelligent Transportation Systems*, *18*(11), 2916–2928.

He, C., Shao, J., Zhang, J., & Zhou, X. (2020). Clustering-based multiple instance learning with multi-view feature. *Expert Systems with Applications*, *162*, Article 113027.

Hotelling, H. (1936). Relations between two sets of variates. *Biometrika*, *28*(3/4), 321–377.

Houthuys, L., Langone, R., & Suykens, J. A. (2018). Multi-view least squares support vector machines classification. *Neurocomputing*, *282*, 78–88.

Jin, R., Wang, S., & Zhou, Z.-H. (2009). Learning a distance metric from multi-instance multi-label data. In *Proceedings of the international conference on computer vision and pattern recognition* (pp. 896–902). IEEE.

Khalilpourazari, S., Doulabi, H. H., Çiftçioğlu, A. O., & Weber, G.-W. (2021). Gradient-based grey wolf optimizer with Gaussian walk: Application in modelling and prediction of the COVID-19 pandemic. *Expert Systems with Applications*, *177*, Article 114920.

Lazebnik, S., Schmid, C., & Ponce, J. (2004). Semi-local affine parts for object recognition. In *British machine vision conference* (pp. 779–788). The British Machine Vision Association (BMVA).

Li, F.-F., & Perona, P. (2005). A Bayesian hierarchical model for learning natural scene categories. In *Proceedings of the international conference on computer vision and pattern recognition* (pp. 524–531). IEEE Computer Society.

Li, D., Xu, D., Tang, J., & Tian, Y. (2017). Metric learning for multi-instance classification with collapsed bags. In *2017 international joint conference on neural networks* (pp. 372–379). IEEE.

Lienhart, R., & Maydt, J. (2002). An extended set of haar-like features for rapid object detection. In *Proceedings of the international conference on image processing* (pp. 1–4). IEEE.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, *60*(2), 91–110.

Maron, O., & Lozano-Pérez, T. (1998). A framework for multiple-instance learning. In *Advances in neural information processing systems* (pp. 570–576).

Melki, G., Cano, A., & Ventura, S. (2018). MIRSVM: Multi-instance support vector machine with bag representatives. *Pattern Recognition*, *79*, 228–241.

Miao, Y., Tao, X., Sun, Y., Li, Y., & Lu, J. (2015). Risk-based adaptive metric learning for nearest neighbour classification. *Neurocomputing*, *156*, 33–41.

Misra, D., Mohanty, S. N., Agarwal, M., & Gupta, S. K. (2020). Convoluted cosmos: classifying galaxy images using deep learning. In *Data management, analytics and innovation* (pp. 569–579). Springer.

Mohanty, R., Mallik, B. K., & Solanki, S. S. (2020). Automatic bird species recognition system using neural network based on spike. *Applied Acoustics*, *161*, Article 107177.

Nan Zhang, H. L., & Jia, W. (2019). Multimodal correlation deep belief networks for multi-view classification. *Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies*, *49*(5), 1925–1936.

Niu, G., Dai, B., Yamada, M., & Sugiyama, M. (2014). Information-theoretic semi-supervised metric learning via entropy regularization. *Neural Computation*, *26*(8), 1717–1762.

Pang, C., Wang, W., Lan, R., Shi, Z., & Luo, X. (2021). Bilinear pyramid network for flower species categorization. *Multimedia Tools and Applications*, *80*(1), 215–225.

Park, K., Shen, C., Hao, Z., & Kim, J. (2011). Efficiently learning a distance metric for large margin nearest neighbor classification. In *Proceedings of the AAAI conference on artificial intelligence*,

Phillips, P. J., Moon, H., Rizvi, S. A., & Rauss, P. J. (2000). The FERET evaluation methodology for face-recognition algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(10), 1090–1104.

Qiu, H., Gong, D., Li, Z., Liu, W., & Tao, D. (2021). End2End occluded face recognition by masking corrupted features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Rakotomamonjy, A., Bach, F. R., Canu, S., & Grandvalet, Y. (2008). Simplemkl. *Journal of Machine Learning Research*, *9*(3), 2491–2521.

Shen, C., Kim, J., Wang, L., & Hengel, A. (2009). Positive semidefinite metric learning with boosting. In *Advances in neural information processing systems* (pp. 1651–1659).

Tang, J., He, Y., Tian, Y., Liu, D., Kou, G., & Alsaadi, F. E. (2021). Coupling loss and self-used privileged information guided multi-view transfer learning. *Information Sciences*, *551*, 245–269.

Tang, J., & Tian, Y. (2017). A multi-kernel framework with nonparallel support vector machine. *Neurocomputing*, *266*, 226–238.

Tang, J., Tian, Y., Liu, D., & Kou, G. (2019). Coupling privileged kernel method for multi-view learning. *Information Sciences*, *481*, 110–127.

Tang, J., Tian, Y., Zhang, P., & Liu, X. (2017). Multiview privileged support vector machines. *IEEE Transactions on Neural Networks and Learning Systems*, *29*(8), 3463–3477.

Tang, J., Xu, W., Li, J., Tian, Y., & Xu, S. (2021). Multi-view learning methods with the LINEX loss for pattern classification. *Knowledge-Based Systems*, *228*, Article 107285.

Torresani, L., & Lee, K.-c. (2006). Large margin component analysis. In *Advances in neural information processing systems* (pp. 1385–1392).

Wang, W., Lai, Q., Fu, H., Shen, J., Ling, H., & Yang, R. (2021). Salient object detection in the deep learning era: An in-depth survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Wang, F., & Sun, J. (2014). Survey on distance metric learning and dimensionality reduction in data mining. *Data Mining and Knowledge Discovery*, *29*(2), 534–564.

Wang, Y., Zhang, W., Wu, L., Lin, X., & Zhao, X. (2017). Unsupervised metric fusion over multiview data by graph random walk-based cross-view diffusion. *IEEE Transactions on Neural Networks and Learning Systems*, *28*(1), 57–70.

Wang, W., & Zhou, Z. (2010). A new analysis of co-training. In *Proceedings of the international conference on machine learning* (pp. 1135–1142).

Wang, J., & Zucker, J. D. (2000). Solving multiple-instance problem: A lazy learning approach. In *Proceedings of the international conference on machine learning* (pp. 1119–1126).

Weinberger, K. Q., Blitzer, J., & Saul, L. K. (2005). Distance metric learning for large margin nearest neighbor classification. In *Advances in neural information processing systems* (pp. 1473–1480).

Weinberger, K. Q., & Saul, L. K. (2009). Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, *10*, 207–244.

Wen, C., Guyer, D. E., & Li, W. (2009). Local feature-based identification and classification for orchard insects. *Biosystems Engineering*, *104*(3), 299–307.

Xiao, Y., Liu, B., Hao, Z., & Cao, L. (2013). A similarity-based classification framework for multiple-instance learning. *IEEE Transactions on Cybernetics*, *44*(4), 500–515.

Xing, E. P., Jordan, M. I., Russell, S., & Ng, A. Y. (2002). Distance metric learning with application to clustering with side-information. In *Advances in neural information processing systems* (pp. 505–512).

Xu, Y., Ping, W., & Campbell, A. T. (2011). Multi-instance metric learning. In *Proceedings of the international conference on data mining* (pp. 874–883). IEEE.

Ying, Y., & Li, P. (2012). Distance metric learning with eigenvalue optimization. *Journal of Machine Learning Research*, *13*(1), 1–26.

Yu, J., Wang, M., & Tao, D. (2012). Semisupervised multiview distance metric learning for cartoon synthesis. *IEEE Transactions on Image Processing*, *21*(11), 4636–4648.

Zhang, Q., & Goldman, S. A. (2001). EM-DD: An improved multiple-instance learning technique. In *Advances in neural information processing systems* (pp. 1073–1080).