Multi-Instance Dimensionality Reduction

Yu-Yin Sun[1] **Michael K. Ng**[2] **Zhi-Hua Zhou**[1*]
[1] National Key Laboratory for Novel Software Technology, Nanjing University, China
[2] Department of Mathematics, Hong Kong Baptist University, Hong Kong, China
sunyy@lamda.nju.edu.cn mng@math.hkbu.edu.hk zhouzh@lamda.nju.edu.cn

Abstract

Multi-instance learning deals with problems that treat bags of instances as training examples. In single-instance learning problems, dimensionality reduction is an essential step for high-dimensional data analysis and has been studied for years. The *curse of dimensionality* also exists in multi-instance learning tasks, yet this difficult task has not been studied before. Direct application of existing single-instance dimensionality reduction objectives to multi-instance learning tasks may not work well since it ignores the characteristic of multi-instance learning that the labels of bags are known while the labels of instances are unknown. In this paper, we propose an effective model and develop an efficient algorithm to solve the multi-instance dimensionality reduction problem. We formulate the objective as an optimization problem by considering orthonormality and sparsity constraints in the projection matrix for dimensionality reduction, and then solve it by the gradient descent along the tangent space of the orthonormal matrices. We also propose an approximation for improving the efficiency. Experimental results validate the effectiveness of the proposed method.

Introduction

In single-instance scenario we are given a training set containing N instances with their labels. In *multi-instance learning* (Dietterich, Lathrop, and Lozano-Perez 1997) the training examples are N *bags* each containing many instances. The labels of training bags are known yet the labels of the training instances are unknown. According to the standard multi-instance learning assumption, a positive bag contains at least one positive instance, while all the instances in negative bags are negative.

Multi-instance learning has been found useful in modeling many real world applications such as drug activity prediction (Dietterich, Lathrop, and Lozano-Perez 1997), image retrieval (Andrews, Tsochantaridis, and Hofmann 2003), text categorization (Andrews, Tsochantaridis, and Hofmann 2003), face detection (Viola, Platt, and Zhang 2006), computer-aided medical diagnosis (Fung et al. 2007), etc. Many of these tasks involve high-dimensional data and thus encounter the *curse of dimensionality*.

In single-instance scenario the curse of dimensionality has attracted much attention. There are two major paradigms, i.e., feature selection and dimensionality reduction. Feature selection tries to select a subset of the original features according to some measurements such as the mutual information or distance-based measures. Raykar et al. (2008) have studied multi-instance feature selection using Bayesian method which automatically considers the feature relevance. In most cases, searching an optimal feature subset is hard and heuristic methods are often used. Dimensionality reduction, which tries to extract a small number of new features by projecting the original features into a new space, is generally with better theoretical foundation. Existing dimensionality reduction techniques can be roughly divided into two categories, that is, unsupervised approaches such as PCA (principal component analysis) (Jolliffe 2002), and supervised approaches such as LDA (linear discriminant analysis) (Fukunaga 1990). To the best of our knowledge, multi-instance dimensionality reduction has not been studied before. It is noteworthy that multi-instance dimensionality reduction is even harder than single-instance dimensionality reduction since the input space of multi-instance learning task is ambiguous.

In this paper, we propose the MIDR (Multi-Instance Dimensionality Reduction) approach based on a specifically designed dimensionality reduction objective for multi-instance learning. We formulate the objective as an optimization problem by considering orthonormality and sparsity constraints in the projection matrix for dimensionality reduction, and then solve it by gradient descent along the tangent space of the orthonormal matrices. We also propose an approximation to improve the efficiency. Experimental results validate the effectiveness of the proposed method.

The rest of this paper is organized as follows. We start by a brief review of related work. Then, we propose MIDR and report our experiments, which is followed by the conclusion.

Related Work

In single instance scenario, we are given a training set $\{(\boldsymbol{x}_1, y_1), \cdots, (\boldsymbol{x}_N, y_N)\}$ where $\boldsymbol{x}_i \in \mathbb{R}^D$ is an instance and $y_i \in \{1, \cdots, k\}$ is its label. To extract $d \ll D$ features, linear dimensionality reduction methods apply a lin-

*Supported by the National Fundamental Research Program of China (2010CB327903), the National Science Foundation of China (60635030, 60721002) and Jiangsu Science Foundation (BK2008018).

ear transformation $A \in \mathbb{R}^{D\times d}$ to project each data point $\boldsymbol{x}_i$ into a lower-dimensional space $\mathbb{R}^d$ as $A^{\mathrm{T}}\boldsymbol{x}_i$.

LDA (Fukunaga 1990), a representative of supervised linear dimensionality reduction method, tries to maximize the between-class distance and minimize the within-class distance at the same time. In order to make the resulting features uncorrelated, A is required to be orthonormal. Thus, the optimization problem of LDA is

$$\max_{A^{\mathrm{T}}A=I_d} \mathrm{tr}\left(\frac{A^{\mathrm{T}}S_bA}{A^{\mathrm{T}}S_wA}\right) ,$$

where S_b and S_w are the between-class and within-class covariance matrix, respectively, and $\mathrm{tr}(\cdot)$ is the matrix trace.

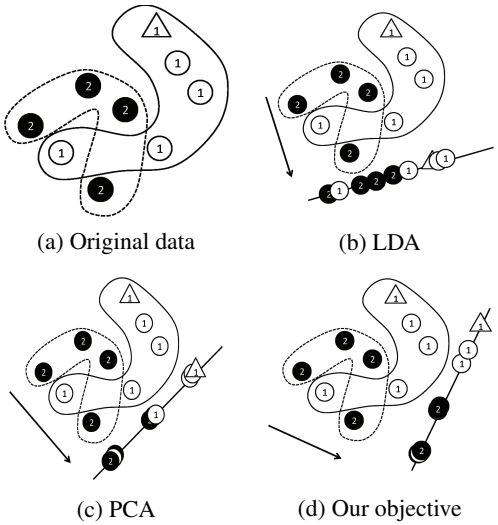
PCA (Jolliffe 2002), a representative of unsupervised linear dimensionality reduction method, tries to maximize the variance of the projected data. With the orthonormality constraint, the optimization problem of PCA is

$$\max_{A^{\mathrm{T}}A=I_d} A^{\mathrm{T}}SA ,$$

where S is the data covariance matrix.

In multi-instance scenario, the training set is $\{(X_1, y_1), \cdots, (X_N, y_N)\}$, where $X_i = \{\boldsymbol{x}_{i1}, \cdots \boldsymbol{x}_{in_i}\} \subseteq \mathbb{R}^D$ is a *bag* and $y_i \in \{0, 1\}$ is the label of X_i, $\boldsymbol{x}_{ij}$ denotes the j^{th} *instance* in the i^{th} bag and its hidden label is $y_{ij} \in \{0, 1\}$. According to the standard multi-instance assumption, if there exists at least one instance $\boldsymbol{x}_{ij} \in X_i$ has label $y_{ij} = 1$, X_i's label $y_i = 1$ and $\boldsymbol{x}_{ij}$ is the key (positive) instance. If all instances $\boldsymbol{x}_{ij} \in X_i$ have label $y_{ij} = 0$, X_i's label $y_i = 0$. The ambiguity of input space makes the direct application of single-instance dimensionality reduction methods to multi-instance tasks improper. For example, to apply LDA to multi-instance problems, we need to assign a label for each instance. One approach is to assign each instance to the label of the bag it belongs to. However, although all the instances in the negative bags are negative, most instances in positive bags are generally not positive. Thus, LDA may be misled by the negative instances in positive bags. Figure 1(b) gives an illustration. LDA tries to push all the instances in positive bags together in order to reduce the "within-class" distance no matter what their potential labels are. This may actually decrease the "between-class" distance between positive and negative instances. To apply PCA, we can treat instances in all the bags as the input. However, PCA does not take label information into account and thus the labels of the bags are ignored. Figure 1(c) illustrates that the direct application of PCA could not result in good performance. Another possibility is to estimate the positive instances in positive bags at first, and then apply single-instance supervised dimensionality reduction methods. However, estimating positive instances in positive bags is a challenging problem, and just recently there are a few studies (Zhou, Xue, and Jiang 2005; Li et al. 2009).

Since single-instance dimensionality reduction methods could not meet the requirement of multi-instance learning problems, we study the dimensionality reduction for multi-instance learning and propose the MIDR method.



(a) Original data (b) LDA (c) PCA (d) Our objective

Figure 1: Illustration of applying single-instance dimensionality reduction methods to multi-instance learning task. Triangles represent positive instances, circles represent negative instances, and the numbers in the triangles/circles indicate the index of the bag where bag 1 is positive while bag 2 is negative. For a better representation, each bag is shown with a contour. (a) the original multi-instance data, (b) and (c) the projection direction and projected results of LDA and PCA, respectively, (d) our objective.

The MIDR Approach

Formulation

Our motivation of multi-instance dimensionality reduction is to learn a projection matrix A and after the projection it is easy to discriminate positive and negative bags. Denote the i^{th} projected bag as $A^{\mathrm{T}}X_i = \{A^{\mathrm{T}}\boldsymbol{x}_{i1}, \cdots, A^{\mathrm{T}}\boldsymbol{x}_{in_i}\}$, and we want its posterior probability of being positive, $\Pr(y_i = 1|A^{\mathrm{T}}X_i)$, to be close to one if it is positive and zero if it is negative. By introducing the squared loss, it is equivalent to solve the optimization problem

$$\min_A \sum\nolimits_i \left(\Pr(y_i = 1|A^{\mathrm{T}}X_i) - y_i\right)^2 . \qquad (1)$$

According to the standard multi-instance learning assumption, i.e., a key (positive) instance decides a bag's label, we can express the posterior probability of a bag in terms of the posterior probabilities of its instances as

$$\Pr(y_i = 1|A^{\mathrm{T}}X_i) = \max_j \Pr\left(y_{ij} = 1|A^{\mathrm{T}}\boldsymbol{x}_{ij}\right) .$$

Then Eq. 1 becomes

$$\min_A \sum\nolimits_i \left(\max_j \Pr\left(y_{ij} = 1|A^{\mathrm{T}}\boldsymbol{x}_{ij}\right) - y_i\right)^2 . \qquad (2)$$

Here we can see that in order to minimize Eq. 2, we should enlarge the distance between the key (positive) instance and negative instances, as illustrated in Figure 1(d).

To make our objective smooth, we replace the max in Eq. 2 by $softmax$ as

$$P_i = \mathrm{softmax}_\alpha(P_{i1}, \cdots, P_{in_i}) = \frac{\sum_j P_{ij}e^{\alpha P_{ij}}}{\sum_j e^{\alpha P_{ij}}} ,$$

where we denote $\Pr(y_i = 1|A^{\mathrm{T}}X_i)$ and $\Pr\left(y_{ij} = 1|A^{\mathrm{T}}\boldsymbol{x}_{ij}\right)$ as P_i and P_{ij} respectively for convenience. α is a parameter controlling the extent to which the softmax approximates the max function.

Similar to single-instance dimensionality reduction methods, we also require the resulting features to be uncorrelated, i.e., we require A to be orthonormal. Thus our optimization problem becomes

$$\min_{A} \quad \sum_i (P_i - y_i)^2 \quad \text{s.t.} \quad A^{\mathrm{T}}A = I_d \; . \tag{3}$$

We also want A to be sparse, and this can be achieved by enforcing the l_1-norm regularization. Thus we attempt to solve the optimization problem

$$\begin{aligned} \min_{A} \quad & \sum_i (P_i - y_i)^2 + C_1 \sum_{s,t} |A_{st}| \\ \text{s.t.} \quad & A^{\mathrm{T}}A = I_d \; , \end{aligned} \tag{4}$$

where $A = [A_{st}]_{D\times d}$ and C_1 is a controlling parameter.

Note that the objective function of Eq. 4 is not smooth because of the additional term $\sum_{s,t} |A_{st}|$. We therefore, approximate this term by

$$|A_{st}| \approx |A_{st}\,(\epsilon)\,| = \sqrt{A_{st}^2 + \epsilon^2}, \; \epsilon > 0 \; ,$$

where ϵ is a small positive constant (Qi and Sun 2000). Our optimization problem consequently becomes

$$\begin{aligned} \min_{A} \quad & \sum_i (P_i - y_i)^2 + C_1 \sum_{s,t} |A_{st}(\epsilon)| \\ \text{s.t.} \quad & A^{\mathrm{T}}A = I_d \; . \end{aligned} \tag{5}$$

Gradient-Descent Search

Since the columns of projection matrix A are constrained to be orthonormal, we consider the problem in a set

$$St(d, D) = \{A \in \mathcal{R}^{D\times d} | A^{\mathrm{T}}A = I_d\}$$

which contains all $D \times d$ matrices with orthonormal columns. Note that $St(d, D)$ is a compact smooth manifold called the *compact Stiefel manifold* (Stiefel 1935), and its tangent space $T_A St(d, D)$ at any $A \in St(d, D)$ can be expressed by (Helmke and Moore 1994)

$$T_A St(d, D) = \{X \in \mathcal{R}^{D\times d} | X^{\mathrm{T}}A + A^{\mathrm{T}}X = 0\}. \tag{6}$$

Regarding the manifold $St(d, D)$ as an embedded submanifold of the Euclidean space, the standard inner product, or the Frobenius inner product for $D \times d$ matrices defined by

$$\langle X, Y\rangle = \text{trace}(X^{\mathrm{T}}Y), \forall\, X, Y \in T_A St(d, D)$$

is induced and referred as the induced Riemannian metric on $St(d, D)$.

The projections of any $Z \in \mathcal{R}^{D\times d}$ onto the tangent space $T_A St\,(d, D)$ at A can be defined as (Chu and Trendafilov 2001)

$$\Pi_T(Z) = A(\frac{A^{\mathrm{T}}Z - Z^{\mathrm{T}}A}{2}) + (I_D - AA^{\mathrm{T}})Z \; . \tag{7}$$

Suppose a smooth function $\phi : St(d, D) \to \mathcal{R}$ is defined on $St(d, D)$ like the objective function in Eq. 5. Then the gradient $\text{grad}(\phi(A))$ of ϕ at $A \in St(d, D)$ is given by (Helmke and Moore 1994; Edelman, Arias, and Smith 1998; Chu and Trendafilov 2001)

$$\text{grad}(\phi(A)) = \Pi_T(\partial\phi(A)/\partial A), \forall A \in St(d, D) \; . \tag{8}$$

We can compute the gradient of our objective function $\partial\phi(A)/\partial A$ explicitly, hence we can easily get the gradient flow in the tangent space from Eq. 8 with the aid of the orthonormal projection $\Pi_T(Z)$ defined by Eq. 7. The gradient flow is an ordinary differential equation for the minimization of $\phi(A)$ in Eq. 5 (Chu and Trendafilov 2001), i.e.,

$$dA(t)/dt = -\text{grad}(\phi(A)) = -\Pi_T(\partial\phi(A)/\partial A) \; . \tag{9}$$

Here the variable t can be interpreted as the time step towards finding the optimal solution of Eq. 5. This is a gradient system based on the optimality conditions of the problem. To get A we can solve an ordinary differential equation problem using routine processes such as the MATLAB ode23 or Maple with the aid of Eq. 9.

Convergence

Since $St(d, D)$ is a compact smooth manifold, and $\phi(A)$ is already a smooth function on $St(d, D)$, the routine convergence results for the preceding gradient system (Helmke and Moore 1994) can be applied.

Moreover, it is worth noting that for any initial value $A(0) = A_0 \in St(d, D)$, there is a unique trajectory $A(t)$ starting from A_0 for $t \geq 0$, and since $\Pi_T\left(\partial\phi(A)/\partial A\right) \in T_A St(d, D)$, it immediately yields by Eq. 7 that

$$d(A(t)^{\mathrm{T}}A(t))/dt = \dot{A}(t)^{\mathrm{T}}A(t) + A(t)^{\mathrm{T}}\dot{A}(t) \equiv 0, \quad t \geq 0 \; ,$$

implying $A(t) \in St(d, D)$ for $t \geq 0$. Here we use $\dot{A}(t) = dA(t)/dt$ for short. Additionally, for any $Z \in \mathbb{R}^{D\times d}$, it follows that $\langle Z, \Pi_T(Z)\rangle = \langle \Pi_T(Z), \Pi_T(Z)\rangle \geq 0$ (Chu and Trendafilov 2001), and hence for any $t \geq 0$,

$$\begin{aligned} \frac{d\phi(A(t))}{dt} &= \langle \frac{\partial\phi(A)}{\partial A}, \Pi_T\left(\frac{\partial\phi(A)}{\partial A}\right)\rangle \\ &= \langle \Pi_T\left(\frac{\partial\phi(A)}{\partial A}\right), \Pi_T\left(\frac{\partial\phi(A)}{\partial A}\right)\rangle \geq 0 \; , \end{aligned}$$

implying the monotonically non-increasing property of the objective function $\phi(A(t))$ along the trajectory $A(t) \in St(d, D)$ for $t \geq 0$.

It should be emphasized here that any local maximum (or local minimum) of the function $\phi(A) : St(d, D) \to \mathbb{R}$ is a critical point, and the gradient flow $A(t)$ defined by Eq. 9 exists for all $t \geq 0$, and converges to a connected component of the set of critical points of $\phi(A)$ as $t \to \infty$. Furthermore, if $\phi(A)$ has only isolated critical points, $A(t)$ is guaranteed to converge to one critical point when $t \to \infty$ (Helmke and Moore 1994).

As the descent direction is used, the objective function value is decreasing at each step. It is clear the lower bound of the objective function is zero. Therefore, the gradient descent process can stop when the objective function value does not improve with respect to the iterations.

Speedup

Solving ordinary differential equations at every iteration is time-consuming especially when the dimensionality is very high. Here we provide an approximation for updating A to improve the efficiency. Consider the new problem

$$\min_{A,H} \sum\nolimits_i (P_i - y_i)^2 + \frac{C_2}{2}\|A - H\|_F^2 + C_1 \sum\nolimits_{s,t} |H_{st}| \quad (10)$$

with constraint $A^{\mathrm{T}}A = I_d$. Here we keep A to be orthonormal, and H to be sparse. Now we need to update A, and then H. To update A, we let

$$\psi(A) = \sum\nolimits_i (P_i - y_i)^2 + \frac{C_2}{2}\|A - H\|_F^2 \,,$$

and use the simple gradient descent formula

$$A_{\text{new}} = A_{\text{old}} - \gamma \text{grad}(\psi(A)) \,. \quad (11)$$

The orthogonality based on the one-step gradient flow can be preserved. A line search procedure for the step size γ can be implemented together. To update H, we can use soft-thresholding. For each (i, j), we solve the minimization problem

$$\min_{H_{ij}} \|A_{ij} - H_{ij}\|_2^2 + 2\frac{C_1}{C_2}|H_{ij}| \,.$$

The optimizer is

$$H_{ij} = \begin{cases} A_{ij} - \frac{C_1}{C_2}, & \text{if } A_{ij} > \frac{C_1}{C_2} \\ 0, & \text{if } -\frac{C_1}{C_2} \le A_{ij} \le \frac{C_1}{C_2} \\ A_{ij} + \frac{C_1}{C_2}, & \text{if } A_{ij} < -\frac{C_1}{C_2} \,. \end{cases} \quad (12)$$

The sparsity is controlled by the parameters C_1 and C_2, and the orthonormality loss of H is controlled by C_2.

The overall approach is summarized in Algorithm 1. H is initialized to be a d-cardinality set of orthonormal vectors. γ is set to 1 at the beginning and at each iteration new γ is updated by line search. The posterior probability P_{ij} and P_i can be estimated in many ways, such as naïve Bayes, SVM, neural networks, logistic regression, etc. Here we denote P_i as a probability estimation function whose parameters are represented by $\boldsymbol{v}$ for convenience. At each iteration, we update the parameters $\boldsymbol{v}$ and the posterior probability estimation function P_i. Note that in our approach we use the true gradient rather than stochastic gradient because the convergence of stochastic gradient approach is usually slower than the true gradient approach. In MIDR the cost of stochastic gradient approach will be even higher since we may need to use a bag of instances instead of a single sample to fit the parameters and need to deal with the orthogonal constraints.

Algorithm 1: The MIDR algorithm

Input: C_1, C_2 and α
Output: The final solution of A
initialize A, H, $\boldsymbol{v}$ and γ;
while *not converge* **do**
1. update $\boldsymbol{v}$ and P_i
2. compute grad$(\psi(A))$
3. $\gamma \leftarrow$linesearch$(\gamma, \text{grad}(\psi(A)))$
4. update A by setting
$$A_{\text{new}} = A_{\text{old}} - \gamma \text{grad}(\psi(A))$$
5. update H based on Eq. 12

end

Experiments

Configuration

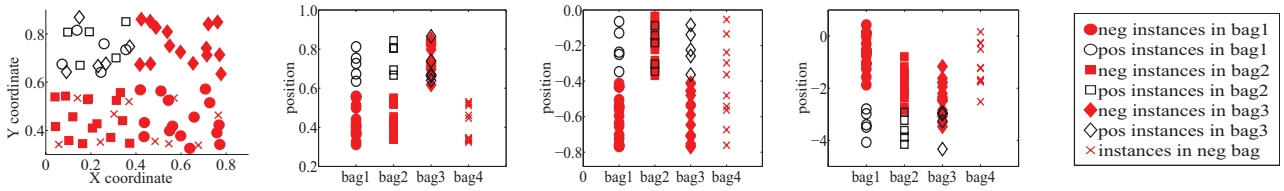
There is no multi-instance dimensionality reduction method before, and so we compare our proposed MIDR approach with several modified single-instance dimensionality reduction methods. We take LDA and PCA for the representatives of supervised and unsupervised dimensionality reduction methods, respectively. LDA should be provided with the labels of all the input data, but in multi-instance learning we only have the labels of training bags. All the instances in negative bags are negative and here we assign positive labels to all the instances in positive bags. As for PCA, we take instances in all the bags as input.

We compare the methods on a synthetic data and five multi-instance benchmark data sets. For the synthetic data, as shown in Figure 2(a), all the dimensionality reduction methods reduce the dimensionality from 2 to 1. This data set is very simple, but is helpful for understanding the behaviors of these methods through visualization. The benchmark data sets used here are *Musk1*, *Musk2*, *elephant*, *fox* and *tiger*. These data sets have been used in most multi-instance learning studies. *Musk1* and *Musk2* are drug activity prediction tasks, where each instance is represented by an 166-dimensional feature vector. Detailed information can be found in (Dietterich, Lathrop, and Lozano-Perez 1997). *Elephant*, *fox* and *tiger* are image classification tasks, where each instance is described by a 230-dimensional feature vector. Detailed information can be found in (Andrews, Tsochantaridis, and Hofmann 2003).

For the benchmark data sets, we compare the methods via 5-fold cross validation (we repeat 10 times 5-fold cross validation with random partitions). For MIDR, the parameter α controlling the softmax is set to the fixed number 3.5 as suggested by (Ray and Craven 2005), and the parameters C_1 and C_2 controlling the regularization is picked from the pool of $\{10^i | i = -4, -3, \cdots, 3, 4\}$ by 5-fold cross validation on the training data. We use logistic model to estimate the posterior probability, which has been used in (Xin and Frank 2004; Ray and Craven 2005). For PCA and MIDR, we have tried to reduce the dimensionality to $(20\%, 30\%, 40\%, 50\%, 60\%)$ of the original dimension. We use multi-instance logistic regression (Ray and Craven 2005) as the classifier to evaluate the classification performance. We also compare with the original multi-instance logistic regression without dimensionality reduction, denoted by ORI, solved by an optimization package L-BFGS (Nocedal and Wright 1999). The evaluation criteria used here is AUROC (area under ROC curve).

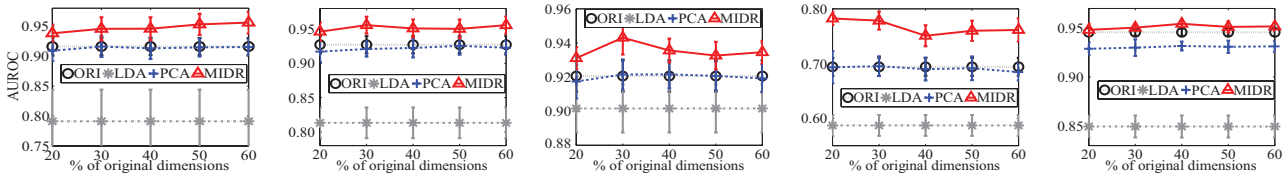
Table 1: Comparison of AUROC (mean±std). Bold values highlight the best AUROC on each data set.

Comparison Method	Data set *Musk1*	*Musk2*	*Elephant*	*Fox*	*Tiger*
ORI	0.916±0.014	0.927±0.013	0.921±0.009	0.694±0.017	0.946±0.004
LDA	0.791±0.053	0.813±0.023	0.902±0.014	0.587±0.019	0.850±0.011
PCA	0.916±0.018	0.921±0.011	0.922±0.009	0.695±0.018	0.930±0.009
MIDR	**0.946±0.019**	**0.955±0.011**	**0.943±0.010**	**0.778±0.017**	**0.950±0.003**



(a) Original data (b) LDA (c) PCA (d) MIDR

Figure 2: The dimensionality reduction results on the synthetic data. (b), (c) and (d) show the location of instances from different bags on the resulted dimension.



(a) *Musk1* (b) *Musk2* (c) *elephant* (d) *fox* (e) *tiger*

Figure 3: The performance on the benchmark data sets when reduced to different dimensionalities.

Results

Results on the synthetic data are shown in Figure 2. From Figure 2(b) it can be seen that LDA is misled by the negative instances in positive bags. The positive and negative instances in the positive bag 3 are very close to each other after dimensionality reduction. In Figure 2(c), the positive and negative bags are not separated when reduced to 1-dimension since PCA ignores the labels of bags. It can be seen from Figure 2(d) that MIDR tries to enlarge the distances between positive and negative instances and thus the positive and negative bags are easier to separate based on the key instance assumption. It is clear that the proposed MIDR method performs better than LDA and PCA.

Results on the benchmark data sets are summarized in Table 1. Here we only show the results of PCA and MIDR when reduced to 30% dimensions (the other results are shown in Figure 3). 30% is almost the best result for PCA on most of the data sets but not for MIDR. The best result is highlighted in bold face. It is clear that MIDR performs better than all the other methods. LDA performs the worst in almost all cases since the ground-truth labels of the instances in positive bags are unknown while LDA was fooled by negative instances in positive bags. For *elephant*, the performance of LDA is not bad. This is not strange; as Andrews et al. (2003) disclosed, supervised learning methods can perform well on *elephant* by assigning the label of a bag to its instances. It can also be seen that the performance of PCA is quite close to the ORI performance. This verifies what we have pointed out before, that is, PCA preserves as much data variance as possible but ignores the label information, thus it is less helpful for multi-instance classification.

We also record the average time costs under different parameter settings of these methods, as shown in Table 2. It can be seen that for most cases, the time cost of MIDR is larger than that of LDA but smaller than that of PCA. LDA seems more efficient partly because that it reduces to one dimension, and the time cost of training multi-instance logistic regression with one-dimensional feature vectors is significantly smaller than using more features.

Figure 3 compares the methods under different dimensionalities to be reduced. MILR always works better than other methods on all dimensions. The performance curve increases smoothly on *Musk1* and *Musk2*, yet the curves do not increase smoothly on the image categorization tasks. This may be caused by the well-known large gap between the low-level image features and high-level image semantics.

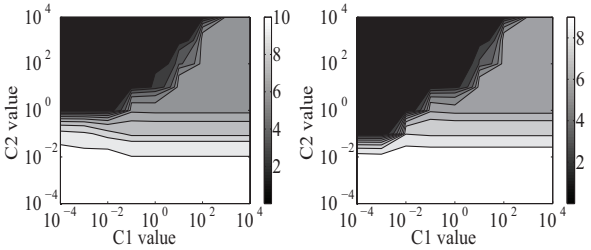
Sparsity and Orthonormality

We empirically study how the orthonormality and sparsity of the linear projection matrix A are affected by different settings of C_1 and C_2. Due to the limit of space, in Figures 4 and 5 we only show the results on *Musk* data sets.

We use $\|H^{\mathrm{T}}H - I_d\|_F$ to measure the *loss of orthonormality*, as shown in Figure 4. It verifies what we have expected, i.e., as C_2 increases A is getting closer to an or-

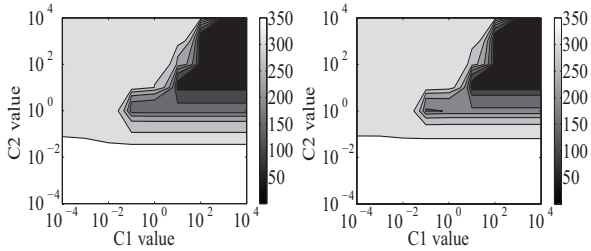
Table 2: Comparison of time costs (mean±std) (in seconds). The time costs include both the dimensionality reduction and classification costs.

Comparison	Data set				
Method	*Musk1*	*Musk2*	*Elephant*	*Fox*	*Tiger*
LDA	32.519±3.335	218.678±12.602	115.271±10.859	112.091±9.965	117.276±10.632
PCA	65.808±5.672	479.371±15.285	223.642±12.561	225.261±13.865	252.229±14.525
MIDR	40.561±4.518	375.952±13.579	194.119±12.945	205.098±11.119	203.515±13.579



(a) Musk1 (b) Musk2

Figure 4: Loss of orthonormality with C_1 and C_2.



(a) Musk1 (b) Musk2

Figure 5: Loss of sparsity with C_1 and C_2.

thonormal matrix. When C_2 is small, the loss of orthonormality is insensitive to the settings of C_1. When C_2 is large, a larger C_1 may result in less orthonormality. We can also see that with a fixed C_1, the loss of orthonormality decreases exponentially as C_2 increases. We measure the *loss of sparsity* by $\sum_{i,j} |H_{ij}|$, as shown in Figure 5. It can be seen that when C_2 is small, the loss of sparsity is large and insensitive to the settings of C_1.

From Figures 4 and 5 we can see that when C_2 is larger than 10^{-2}, the losses of orthonormality and sparsity are small. There is a trade-off between the orthonormality and sparsity of A, controlled by the setting of C_1.

Conclusion

In this paper, we study the problem of multi-instance dimensionality reduction. We formulate the objective as an optimization problem with orthonormality and sparsity constraints, and propose a gradient descent method in the tangent space to solve this optimization problem. A speed-up method is provided to improve the efficiency. Experimental results show that our method produces encouraging results.

Currently we focus on the standard multi-instance learning assumption, i.e., a key (positive) instance makes a bag positive. There are alternative multi-instance learning assumptions that have been found useful in practice (Foulds and Frank 2009). Studying multi-instance dimensionality reduction in those scenarios is an interesting future issue.

References

Andrews, S.; Tsochantaridis, I.; and Hofmann, T. 2003. Support vector machines for multiple-instance learning. In *NIPS 15*. 561–568.

Chu, M. T., and Trendafilov, N. T. 2001. The orthogonally constrained regression revisited. *Journal of Computational and Graphical Statistics* 10:746–771.

Dietterich, T. G.; Lathrop, R. H.; and Lozano-Perez, T. 1997. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence* 89:31–71.

Edelman, A.; Arias, T. A.; and Smith, S. T. 1998. The geometry of algorithms with orthogonality constraints. *SIAM Journal on Matrix Analysis and Applications* 20:303–353.

Foulds, J., and Frank, E. 2009. A review of multi-instance learning assumptions. *Knowledge Engineering Review*. in press.

Fukunaga, K. 1990. *Introduction to Statistical Pattern Recognition*. Academic Press.

Fung, G.; Dundar, M.; Krishnappuram, B.; and Rao, R. B. 2007. Multiple instance learning for computer aided diagnosis. In *NIPS 19*. 425–432.

Helmke, U., and Moore, J. B. 1994. *Optimization and Dynamical systems*. Springer.

Jolliffe, I. T. 2002. *Principal Component Analysis*. Springer.

Li, Y.-F.; Kwok, J. T.; Tsang, I. W.; and Zhou, Z.-H. 2009. A convex method for locating regions of interest with multi-instance learning. In *ECML PKDD*, 15–30.

Nocedal, J., and Wright, S. J. 1999. *Numerical Optimization*. Springer.

Qi, L., and Sun, D. 2000. Improving the convergence of non-interior point algorithms for nonlinear complementarity problems. *Mathematics of Computation* 69:283–304.

Ray, S., and Craven, M. 2005. Supervised versus multiple instance learning: An empirical comparison. In *ICML*, 697–704.

Raykar, V. C.; Krishnapuram, B.; Bi, J.; Dundar, M.; and Rao, R. B. 2008. Bayesian multiple instance learning: Automatic feature selection and inductive transfer. In *ICML*, 808–815.

Stiefel, E. 1935. Richtungsfelder und fernparallelismus in n-dimensionalel manning faltigkeiten. *Comentarii Mathematici Helvetici* 8:305–353.

Viola, P.; Platt, J.; and Zhang, C. 2006. Multiple instance boosting for object detection. In *NIPS 18*. 1419–1426.

Xin, X., and Frank, E. 2004. Logistic regression and boosting for labeled bags of instances. In *PAKDD*, 272–281.

Zhou, Z.-H.; Xue, X.-B.; and Jiang, Y. 2005. Locating regions of interest in CBIR with multi-instance learning techniques. In *AJCAI*, 92–101.