Learning Instance Specific Distances Using Metric Propagation

De-Chuan Zhan ZHANDC@LAMDA.NJU.EDU.CN
Ming Li LIM@LAMDA.NJU.EDU.CN
Yu-Feng Li LIYF@LAMDA.NJU.EDU.CN
Zhi-Hua Zhou ZHOUZH@LAMDA.NJU.EDU.CN
National Key Laboratory for Novel Software Technology Nanjing University, Nanjing 210093, China

Abstract

In many real-world applications, such as image retrieval, it would be natural to measure the distances from one instance to others using *instance specific distance* which captures the distinctions from the perspective of the concerned instance. However, there is no complete framework for learning instance specific distances since existing methods are incapable of learning such distances for test instance and unlabeled data. In this paper, we propose the ISD method to address this issue. The key of ISD is *metric propagation*, that is, propagating and adapting metrics of individual labeled examples to individual unlabeled instances. We formulate the problem into a convex optimization framework and derive efficient solutions. Experiments show that ISD can effectively learn instance specific distances for labeled as well as unlabeled instances. The metric propagation scheme can also be used in other scenarios.

1. Introduction

In many real-world applications, instances may be similar or dissimilar to others for different reasons based on their own characteristics. For example, in image retrieval, a "sky" image may be close to other "sky" images according to distances computed with color features, while a "fishing net" image may be close to other images containing "nets" according to distances computed with texture features. Likewise, in collaborative filtering, even if three users X, Y and Z have similar historical profiles over all items, X would regard Y closer to itself than Z when X and Y are fans of certain types of items. Therefore, instead of applying a *uniform* distance metric for *every* instance, it is more natural to enable each instance to have its own *instance specific distance* for measuring its closeness to other instances from its own perspective.

Actually, in content-based image retrieval there has been a study which tries to compute query-sensitive similarities (Zhou & Dai, 2006). In that method, the similarities among different images are decided after receiving the query image, and the similarities between the same pair of images may be different given different queries. It has been shown that the query-sensitive similarity is effective in image retrieval; that method, however, is based on pure heuristics.

An effective way to obtain the desired distance is to learn a distance function that satisfies some pairwise constraints defined over pairs of instances; this is the main purpose of *metric learning* (Yang, 2006). The constraints generally convey "side information" which specifies whether a pair of instances should be close to (or far from) each other. Such side information can be obtained by consulting the user or comparing labels of instances. Besides pairwise constraints, other kinds of constraints, such as the ones which encode the relationship among triplets, can also be used. Previous studies on metric learning generally focused on learning a uniform Mahalanobis distance for all instances, while only a few studies were devoted to learning different distance functions for different instances.

Frome et al. (2006) proposed a method for learning distinctive distance functions for different instances, by enforcing the distances from the concerned instance to instances with the same label be smaller than that to instances with other labels. Later, Frome et al. (2007) extended this work to enable the comparison between distances computed based on different individual instances. It is noteworthy that both methods can only deal with labeled instances since the label of the concerned instance is involved in the learn-

Appearing in *Proceedings of the 26th International Conference on Machine Learning*, Montreal, Canada, 2009.

ing process. There are two important issues unsolved. First, given a test instance, since its label is unknown, there is no instance specific distance to use. Second, when there are abundant unlabeled data, e.g., in semi-supervised or transductive settings, it is not known how to derive instance specific distances for unlabeled instances. Thus, there is no complete framework for learning instance specific distances, and the usefulness of existing methods is limited in real tasks.

In this paper, we address these two issues by proposing the ISD (Instance Specific Distance) method which works in transductive setting. The key of ISD is *metric propagation*, that is, propagating and adapting the distances learned for individual labeled examples to individual unlabeled instances. To the best of our knowledge, this is the first study on metric propagation, and our idea can also be applied to other scenarios. We formulate the problem into a convex optimization framework and derive efficient solutions. Experiments show that ISD can effectively learn instance specific distances for labeled as well as unlabeled instances.

The rest of this paper is organized as follows. Section 2 briefly reviews some related work. Section 3 proposes the ISD method. Section 4 reports on our experimental results. Finally, Section 5 concludes.

2. Related Work

Metric learning attempts to learn an appropriate distance metric that reflects the underlying relationship between instances. Generally, some pairwise constraints defined over pairs of instances are given by user or induced from labeled data. These constraints, or called "side information", are then used to guide the learning process. Such information can be exploited either globally (Xing et al., 2002; Kwok & Tsang, 2003) or locally (Goldberger et al., 2005; Weinberger et al., 2005). Note that in addition to pairwise constraints, other kinds of constraints can also be used.

Most of previous metric learning studies focused on generating a uniform distance function for all instances, neglecting the fact that different instances may hold different properties. Recently there are several works try to learn different distance functions for different instances. Frome et al. (2006) constructs, for each labeled instance $\mathbf{x}_j$, a distance function $D_j(\mathbf{x}_i)$ which outputs the distance *from* the concerned instance $\mathbf{x}_j$ to another instance $\mathbf{x}_i$. Such distance functions are then optimized separately under a set of constraints requiring that distances from the concerned instance to other instances with different labels must be larger than distances from the concerned instance to other instances with the same label. Considering that the constraints do not contain information shared by other distance functions, the output values of different distance functions are not directly comparable. Hence, a meta learning, which aggregates these distance functions, is further required for the final prediction. Later, Frome et al. (2007) extended the method by enforcing the consistency among the set of distance functions. In addition to the constraints used in (Frome et al., 2006), some "inversed" constraints were specified; that is, the distance from an instance to the concerned instance with the same label should be smaller than that from an instance with some other label to the concerned instance. Thus, by incorporating the interactions among the constraints, they enabled the outputs of the resulting distance functions to be comparable.

It is noteworthy that Frome et al. (2006; 2007) can only generate instance specific distances for labeled examples since the label of the concerned instance is needed for identifying the instances with either the same label or different labels for constraints construction. Given a test instance, instead of learning its instance specific distance, Frome et al. (2006; 2007) used the distance function of each labeled training example to derive a probability for the test instance to have a class label as same as the training example, and then aggregated the probabilities derived from all labeled examples and picked the class with the largest probability for the final prediction. To the best of our knowledge, there is no existing method which is capable of learning an instance specific distance for instances without label information, although this is very needed for establishing a complete framework.

Label propagation is a popular technique. In many graph-based semi-supervised learning approaches (Belkin et al., 2006; Zhu et al., 2003; Zhou et al., 2003), a graph defined over both labeled and unlabeled instances is provided, and the labels are then propagated from labeled instances to unlabeled ones across the graph. In fact, given a graph reflecting the underlying structure of the data, other properties of the data can also be propagated. Recently, Li et al. (2008) tried to propagate pairwise constraints over a predefined graph. In this paper, given such a graph, we propose *metric propagation* for propagating distance metrics from individual labeled examples to individual unlabeled instances. It is possible that metric propagation can also be used in other scenarios.

3. The Proposed Method

We restrict our discussion in a transductive setting, where n labeled examples denoted as $\{\mathbf{x}_i, y_i\}_{i=1}^n$, $\mathbf{x}_i \in$

$\mathcal{R}^d$, $y_i \in \mathcal{Z}$ as well as u unlabeled instances denoted as $\{\mathbf{x}_i\}_{i=n+1}^{n+u}$ are given. Besides, a weight matrix $\mathbf{G}$ describing the relationship between all pairs of instances, no matter labeled or unlabeled, is also provided. According to Zhu et al. (2003), the relationship defined via $\mathbf{G}$ reflects the underlying structure of the data.

Our goal is to learn globally consistent instance specific distance functions $D_i(\mathbf{x}) = \mathbf{w}_i^\top \delta_{\mathbf{x}_i,\mathbf{x}}$ for each instance $\mathbf{x}_i$ $(i = 1, \cdots, n+u)$, where $\delta_{\mathbf{x}_i,\mathbf{x}_j} = (\mathbf{x}_i - \mathbf{x}_j) \odot (\mathbf{x}_i - \mathbf{x}_j)$, $\odot$ is the element-wise product on two vectors. As mentioned before, while one can easily learn instance specific distance for a labeled example based on the information induced from its label, learning instance specific distance function for an unlabeled instance is not straightforward since there is no direct side information available.

By assuming that similar instances share similar properties, the distribution of the instance specific distance functions should be *smooth* within a local area. Given the weight matrix $\mathbf{G}$ which essentially represents the underlying structure of the instances, we can propagate the learned instance specific distance metric from the labeled examples to unlabeled ones by enforcing the smoothness of instance specific distances over the graph during the propagation. Here, we refer to this approach as *metric propagation*, in analogy with *label propagation* in graph-based semi-supervised learning (Belkin et al., 2006; Zhu et al., 2003; Zhou et al., 2003). Note that since the distances are able to interact with each other during the metric propagation, the learned distances are intrinsically consistent.

Instead of explicitly conducting metric propagation while learning the distances for labeled examples, we formulate the metric propagation within a regularized framework which conducts the propagation implicitly by optimizing the regularized objective function

$$\begin{aligned} \min_{\mathbf{W}} \quad & \lambda \sum\nolimits_{i=1}^{n} \sum\nolimits_{j \in \mathcal{S}_i} \ell\left(\hat{y}_{ij}, D_i(\mathbf{x}_j)\right) + \Omega(\mathbf{W}, \mathbf{G}) \\ s.t. \quad & \mathbf{w}_i \geq 0, i = 1, \cdots, n+u, \end{aligned} \tag{1}$$

where $\mathbf{W} = [\mathbf{w}_1, \cdots, \mathbf{w}_{n+u}]$ consists of parameters to be learned for $n+u$ instance specific distance metrics of both labeled examples and unlabeled instances; $\hat{y}_{ij} = 1$ iff $y_i = y_j$, and $\hat{y}_{ij} = -1$ otherwise. ℓ is a convex loss function, such as hinge loss in classification or least square loss in regression. Ω is a regularization term responsible for the implicit metric propagation. The set $\mathcal{S}_i$, induced by the labels of instances, provides the side information with respect to $\mathbf{x}_i$. λ is a regularization parameter. Here, as suggested by Frome et al. (2006; 2007), we enforce $\mathbf{w}_i \geq 0$ to ensure that the distances are non-negative.

Inspired by Zhu et al. (2003), we pack the metric propagation mechanism into regularization term Ω:

$$\Omega(\mathbf{W}, \mathbf{G}) = \sum_{i,j=1}^{n+u} E_{ij} ||\mathbf{w}_i - \mathbf{w}_j||^2 = 2tr(\mathbf{W}^\top \mathbf{L} \mathbf{W}). \tag{2}$$

Here, $\mathbf{E} = \mathbf{U}^{-\frac{1}{2}} \mathbf{G} \mathbf{U}^{-\frac{1}{2}}$ is a normalized weight matrix of $\mathbf{G}$. $\mathbf{G}$ describes pairwise relationship between instances, which is an implement of the graph. In this paper, we assume that $\mathbf{G}$ is given. $\mathbf{U}$ is a diagonal matrix whose (i,i)-entry is the i-th row/column sum of $\mathbf{G}$. $\mathbf{L} = (\mathbf{I} - \mathbf{E})$ is the graph Laplacian and $tr(\mathbf{M})$ denotes the trace of matrix $\mathbf{M}$. Obviously, the minimization of Eq. 2 yields a smooth propagation of the instance specific distances over the graph. Here, the weight matrix $\mathbf{G}$ is firstly given as the Heat kernel with Euclidean distance (Zhu et al., 2003). Furthermore, the $\mathbf{G}$ can be updated with the new instance specific distances induced from ISD. In order to investigate whether ISD can be used to refine the graph construction or not, more details during updating $\mathbf{G}$ will be studied in Section 4.

The side information provided by two instances with the same label or different labels could be sufficient for disclosing which instances should be close to or far from the concerned instance. For simplicity, we follow the methods of utilizing label data in many literatures (Yang, 2006), i.e., we only consider the side information between the concerned instance and one labeled example.

Based on such pairwise side information, we instantiate the loss functions ℓ with L1-Loss and L2-Loss, respectively. We will discuss the solutions to the objective function in Eq. 1 with respect to each of the instantiation in the following subsections. We will show that both of the two solutions are effective and the solution with L2-Loss is more efficient.

As mentioned before, the constraints are not restricted to be pairwise side information. So, the loss function ℓ in ISD is not limited to measure the pairwise relationship between the concerned instance $\mathbf{x}_i$ and one labeled example. More labeled examples can be considered for higher-order information in ISD, just like that in Frome et al. (2006; 2007). If we set $\mathbf{E}$ to be the identity matrix and utilize the L1-Loss based on the "triplet" information induced by two labeled examples other than the concerned instance, our ISD becomes equivalent to (Frome et al., 2006).

3.1. ISD with L1-Loss

We define the loss function ℓ in Eq. 1 based on L1-Loss:

$$\ell\left(\hat{y}_{ij}, D_i(\mathbf{x}_j)\right) = \max\left(0, \hat{y}_{ij}(D_i(\mathbf{x}_j) - \eta)\right), \tag{3}$$

where η is a threshold. According to Eq. 3, $\hat{y}_{ij} = 1$ results in $D_i(\mathbf{x}_j) \leq \eta$ while $\hat{y}_{ij} = -1$ leads to $D_i(\mathbf{x}_j) \geq \eta$. Without loss of generality, we simply set $\eta = 1$.

By introducing slack variables and plugging Eq. 2 and Eq. 3 into Eq. 1, we obtain the following convex optimization problem:

$$\begin{aligned} \min_{\mathbf{W},\ \xi_{i,j}} \quad & \lambda \sum\nolimits_{i=1}^{n} \sum\nolimits_{j \in \mathcal{S}_i} \xi_{i,j} + 2tr(\mathbf{W}^\top \mathbf{L}\mathbf{W}) \\ \text{s.t} \quad & \hat{y}_{ij}(\mathbf{w}_i \delta_{\mathbf{x}_i,\mathbf{x}_j} - 1) \leq \xi_{i,j}, i = 1, \cdots, n \\ & \boldsymbol{\xi} \geq \mathbf{0},\ \mathbf{w}_i \geq 0, i = 1, \cdots, n+u. \end{aligned} \tag{4}$$

Optimizing Eq. 4 with respect to all $\mathbf{w}_i$'s simultaneously is of great computational challenge. Instead of solving it directly, we employ the alternating descent method to solve Eq. 4. The main idea is to sequentially solve one $\mathbf{w}_i$ at each time by fixing the other $\mathbf{w}_j$'s, $j \neq i$. We repeat this procedure until Eq. 4 converges or a maximum number of iteration T is reached.

In each iteration, solving a specific $\mathbf{w}_i$ while fixing other $\mathbf{w}_j$'s ($j \neq i$) yields a standard QP problem:

$$\begin{aligned} \min_{\mathbf{w}_i,\ \xi_j} \quad & \sum\nolimits_{i,j=1}^{n+u} E_{ij}(\mathbf{w}_i^\top \mathbf{w}_i - 2\mathbf{w}_i^\top \mathbf{w}_j) + \lambda \sum\nolimits_{j \in \mathcal{C}_i} \xi_j \\ s.t. \quad & \hat{y}_{ij}(\mathbf{w}_i^\top \delta_{\mathbf{x}_i,\mathbf{x}_j} - 1) \leq \xi_j, \quad j \in \mathcal{C}_i \\ & \boldsymbol{\xi} \geq 0,\ \mathbf{w}_i \geq 0. \end{aligned} \tag{5}$$

Instead of using constraints constructed from *all* labeled examples in Eq. 4, here we only use a subset of these constraints. The subset $\mathcal{C}_i$ is consisted of two parts, i.e., all the inequalities generated from instances with different labels from $\mathbf{x}_i$, and equalities generated from the neighbors of $\mathbf{x}_i$ that have the same label as $\mathbf{x}_i$. The consideration behind this particular setting is that, instances from different classes are usually far from each other while instances from the same class are not necessarily close to each other, e.g., instances with the same label may belong to different clusters which may be scattered in the instance space.

By introducing lagrange multipliers for Eq. 5, we get

$$\begin{aligned} L(\mathbf{w}_i, \boldsymbol{\alpha}, \boldsymbol{\gamma}, \boldsymbol{\xi}, \boldsymbol{\beta}) = & \sum\nolimits_{i,j=1}^{n+u} E_{ij}(\mathbf{w}_i^\top \mathbf{w}_i - 2\mathbf{w}_i^\top \mathbf{w}_j) \\ + \lambda \mathbf{1}^\top \boldsymbol{\xi} + \boldsymbol{\alpha}^\top (\mathbf{Y}_i(\mathbf{D}_{i.}^\top \mathbf{w}_i - 1) - \boldsymbol{\xi}) - \boldsymbol{\beta}^\top \boldsymbol{\xi} - \boldsymbol{\gamma}^\top \mathbf{w}_i\ . \end{aligned}$$

Take the derivative of the Lagrangian with respect to $\mathbf{w}_i$, with KKT condition we get

$$\frac{\partial L}{\partial \mathbf{w}_i} = 2 \sum_{j=1}^{n+u} E_{ij}\mathbf{w}_i - 2 \sum_{j=1}^{n+u} E_{ij}\mathbf{w}_j + \hat{\mathbf{D}}_i \boldsymbol{\alpha} - \boldsymbol{\gamma} = 0.$$

Thus,

$$\mathbf{w}_i = (\boldsymbol{C}_i - \hat{\mathbf{D}}_i \boldsymbol{\alpha}/2 + \boldsymbol{\gamma}/2)/\theta_i, \tag{6}$$

where $\hat{\mathbf{D}}_i = \mathbf{D}_{i.}\mathbf{Y}_i$, $\boldsymbol{C}_i = \sum_j E_{ij}\mathbf{w}_j$, $\theta_i = \sum_j E_{ij}$, $\mathbf{Y}_i$ is a diagonal matrix whose (k,k)-entry is set to $\hat{y}_{ij}$ if $\hat{y}_{ij}$ corresponds to the k-th constraint appearing in Eq. 5. $\boldsymbol{\alpha} \in \mathcal{R}^p$ are the dual variables and $\mathbf{D}_{i.} = \left[\delta_{\mathbf{x}_i,\mathbf{x}_1}, \delta_{\mathbf{x}_i,\mathbf{x}_2}, \cdots, \delta_{\mathbf{x}_i,\mathbf{x}_p}\right] \in \mathcal{R}^{d \times p}$. $p = \text{card}\,(\mathcal{C}_i)$ is the number of constraints.

Substituting Eq. 6 back to the Lagrangian yields the following dual problem for Eq. 5.

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & (\hat{\mathbf{D}}_i \boldsymbol{\alpha} - \boldsymbol{\gamma})^\top (\hat{\mathbf{D}}_i \boldsymbol{\alpha} - \boldsymbol{\gamma}) \\ - \quad & 4(\hat{\mathbf{D}}_i \boldsymbol{\alpha} - \boldsymbol{\gamma})^\top \boldsymbol{C}_i + 4\theta_i \boldsymbol{\alpha}^\top \boldsymbol{y}_{i.} \\ s.t. \quad & 0 \leq \boldsymbol{\alpha} \leq \lambda, \boldsymbol{\gamma} \geq 0. \end{aligned} \tag{7}$$

Since there are less constraints in the dual problem, we choose to solve this dual form of the problem in order to reduce the computational cost. Note that Eq. 7 is also a standard QP problem and the global optimal of $\boldsymbol{\alpha}$ can be effectively found. After that, $\mathbf{w}_i$ is computed by Eq. 6, and hence, all the instance specific distances of both labeled and unlabeled $\mathbf{W}$ can be obtained by iteratively solving the dual problem in Eq. 7. Considering that the loss function in Eq. 4 is an L1-Loss, we refer to this version of ISD as ISD-L1.

3.2. ISD with L2-Loss

Recall that in order to efficiently solve ISD-L1, we employ the alternating descent method to solve the problem in Eq. 4 and replace $\mathcal{S}_i$ with $\mathcal{C}_i$. However, there may still exist many inequality constraints induced from labeled examples whose labels are different from the concerned instance, which will increase the learning time.

Inspired by ν-SVM, we can obtain a more efficient method if the loss ℓ is defined with the L2-Loss:

$$\ell\left(\hat{y}_{i,j}, D_i(\mathbf{x}_j)\right) = \max\left(0, \hat{y}_{i,j}(D_i(\mathbf{x}_j) - \eta)\right)^2. \tag{8}$$

By introducing slack variables and plugging Eq. 2 and Eq. 8 into Eq. 1, we obtain the primal form:

$$\begin{aligned} \min_{\mathbf{W},\xi_{i,j},\rho} \quad & \lambda \sum\nolimits_{i=1}^{n} \sum\nolimits_{j \in \mathcal{C}_i} \xi_{i,j}^2 + 2tr(\mathbf{W}^\top \mathbf{L}\mathbf{W}) - \rho \\ \text{s.t} \quad & \hat{y}_{ij}(\mathbf{w}_i \delta_{\mathbf{x}_i,\mathbf{x}_j} - 1) \leq \xi_{i,j} - \rho,\ \mathbf{w}_i \geq 0. \end{aligned} \tag{9}$$

We first drop the last constraint, i.e., $\mathbf{w}_i \geq 0$, so that the alternating descent method can be used to sequentially solve the Eq. 9, and then we project the solution back to the feasible region. After such simplification, the dual problem for solving the sub-optimization problem with respect to $\mathbf{w}_i$ becomes:

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \boldsymbol{\alpha}^\top (\hat{\mathbf{D}}_i^\top \hat{\mathbf{D}}_i + \frac{\theta_i}{\lambda}\mathbf{I})\boldsymbol{\alpha} + 4(\theta_i \boldsymbol{y}_i^\top - \boldsymbol{C}_i^\top \hat{\mathbf{D}}_i)\boldsymbol{\alpha} \\ s.t. \quad & \boldsymbol{\alpha}^\top \mathbf{1} = 1,\ \boldsymbol{\alpha} \geq 0, \end{aligned} \tag{10}$$

where $\mathbf{I}$ is the identity matrix and $\mathbf{1}$ is a all-one vector.

Note that the Lagrange multipliers in Eq. 10 must satisfy a linear equality (the first constraint), so we can efficiently solve this dual variable $\boldsymbol{\alpha}$ using Sequential Minimal Optimization, where two Lagrange multipliers are selected sequentially for joint optimization.

After the dual variable $\boldsymbol{\alpha}$ is obtained, the primal variable $\mathbf{w}_i$ can be calculated by Eq. 11, which projects the solution to the primal variable into the feasible region defined by the last constraint in Eq. 9.

$$\mathbf{w}_i = \max(0, (\mathbf{C}_i - \hat{\mathbf{D}}_i\boldsymbol{\alpha}/2)/\theta_i). \tag{11}$$

Since the loss function in Eq. 8 is an L2-Loss, we refer to this version of ISD as ISD-L2. By instantiating the loss function ℓ with L2-Loss and solving Eq. 10 in an SMO fashion, the optimization problem can be solved more efficiently than by using L1-Loss as the loss function and solving Eq. 7 via the alternating descent.

4. Experiments

We evaluate the ISD approach on fifteen UCI data sets (Blake et al., 1998) and a COREL image data set. The names of data sets are abbreviated in Table 1. For the COREL image data set, according to (Zhang et al., 2005), 20 image classes are considered and 100 images are selected for each class. One hundred and forty-four visual features are extracted for each image.

For each data set, we randomly sample 2/3 instances to create the labeled training set while the remaining 1/3 instances are used for testing. The distributions of both training set and test set are kept as same as that of the original data set. Recall that ISD works in a transductive setting. Thus, we provide the learner with the test instances whose labels are withheld.

We compare the two versions of the proposed ISD method, i.e., ISD-L1 and ISD-L2, with four state-of-the-art metric learning methods. FSM (Frome et al., 2006) learns instance specific distance for each labeled example. FSSM (Frome et al., 2007) is an extension of FSM where the learned distances of different instances are of global consistency. Both FSM and FSSM are essentially not capable of learning the instance specific distances for the unlabeled instances. Here, the distance of an unlabeled instance with the distances of its neighbors are derived as suggested by Frome et al. (2006; 2007). The other two compared methods learn a uniform Mahalanobis distance for all instances. LMNN (Weinberger et al., 2005) learns a Mahalanobis distance metric such that the k-nearest neighbors always belong to the same class while the instances from different classes are separated by a large margin. DNE (Zhang et al., 2007) learns a Mahalanobis distance metric via low-dimensional embedding to squeeze the instances with the same labels while push away those with different labels within a neighborhood. In addition, we also evaluate the original Euclidian distance, denoted as EUCLID, as the baseline.

In the experiments we fix T, the maximum number of iterations for alternating descent, to five and select the regularization parameter λ from $\{10, 100, 1000, 10000\}$ via five-fold cross validation on training sets. The parameters of the compared methods are set according to the suggestions in (Weinberger et al., 2005; Frome et al., 2006; Frome et al., 2007; Zhang et al., 2007). FSM and FSSM were designed for visual recognition where the constraints used in learning can be easily pruned according to the "feature-to-set" distances. For general data sets, we select the constraints as follows: For an instance $\mathbf{x}_j$, we order all the other labeled examples based on the distances computed from the values of a concerned feature, and then generate constraints using the N nearest neighbors. We take $N = 5$ as suggested in (Frome et al., 2007).

Distance is essential to many real applications, and the learned distances can be evaluated in different scenarios. As an implementation, we plug the learned distances into a k-nearest neighbor classifier, and evaluate the quality of the learned distances based on the classification error rates. Here, the k value is selected from $\{3, 5, 7, 9, 11\}$ via five-fold cross validation.

Note that most of the compared methods could not handle missing values. To make a fair comparison, we fill in the missing values for all the data sets. For numerical features, we fill in the mean value of the concerned feature; for nominal features, we fill in the mode of the concerned feature. We split all the nominal features into a set of binary features to facilitate distance computation. All features are normalized into $[0, 1]$.

We repeat the experiments on each data set for 30 runs with random partitions of training/test instances. The average classification error rates and the corresponding standard deviations are tabulated in Table 1. The best performance for each row is marked by a star, and the *significantly best performances* (Zhou & Yang, 2005) are boldfaced. To identify the significantly best performance, we first compare the other methods with the one of the best performance in terms of the paired t-tests at 95% significance level, and then, the performances of the best method as well as the methods which are not significantly worse than the best method are regarded as significantly best performances. Note that some entries are marked by "N/A"; in most of the

Table 1. Comparison of test error rates (mean ± std.). The best performance on each data set is highlighted by '*'. The performances without significant difference with the best performance are bolded (paired t-tests at 95% significance level).

Dataset	ISD-L1	ISD-L2	EUCLID	DNE	LMNN	FSM	FSSM
anneal	**.051±.011***	.068±.011	.064±.012	.086±.018	.182 ±.016	.105±.091	.109±.016
audiolo	**.074±.040**	**.072 ±.033***	**.073±.035**	**.077±.035**	**.075 ±.029**	.131 ±.032	.134±.029
austral	.161±.019	**.149±.019***	.170 ±.023	.324±.024	.160±.018	.276±.026	.216±.026
autos	**.474±.038***	**.480±.046**	.494±.048	**.481±.043**	N/A	.579±.012	.564 ±.041
balance	**.114±.013**	**.116±.014**	.124±.013	.149±.020	**.113±.012***	.134 ±.020	.143 ±.013
breastw	**.031 ±.010**	**.030±.010***	.033±.010	**.031±.011**	**.031±.010**	.102±.041	.112±.029
clean1	**.236±.037***	.276±.028	.248±.034	.272±.035	.246±.038	N/A	.365±.002
diabete	.287±.019	**.269±.023***	.298±.018	**.275±.029**	.279 ±.031	.342±.050	.322±.232
echocar	**.175±.034***	.189±.035	.200±.044	.194±.043	.209±.050	.198±.036	.193±.026
german	**.277±.015**	**.274±.013***	**.277±.016**	.309±.020	.286±.018	**.275±.021**	**.275 ±.060**
haberma	**.277±.029**	**.273±.025***	**.276±.024**	.287±.034	.291±.030	**.276±.032**	**.276±.029**
heart-s	**.181±.023***	.219±.030	.201±.035	.202±.037	.203±.030	.277±.032	.252±.054
house-v	**.072±.017***	**.076 ±.019**	.083±.019	.143±.022	**.072±.023**	.202±.041	.224±.034
ionosph	.169 ±.029	**.159±.031***	.176±.037	.172±.022	.169±.028	.219±.045	.260 ±.037
spectf	.288±.033	.285±.036	.287±.037	.298 ±.039	.282±.038	**.280 ±.009**	**.272±.007** *
corel	**.681 ±.014**	**.682±.004**	**.683±.016**	.697±.017	**.677±.021***	N/A	N/A

cases, this means that the method fails to return any result within a reasonable response time (i.e., 24 hours for a single training in our case). The only exception is that of LMNN on *autos*; the program of LMNN always quits with some error messages on *autos*. One possible explanation is that *autos* is a multi-class data set, and the lack of training instances of the same class make it difficult to find neighbors of the same class or neighbors from other classes.

Table 1 shows that ISD-L1 has achieved the significantly best performance on 12 data sets while ISD-L2 has on 11 data sets. It is obvious that ISD-L1 and ISD-L2 are among the best of all the compared algorithms. Table 1 also discloses that ISD-L1 and ISD-L2 obtain the lowest error rate on 6 and 7 data sets, respectively, while LMNN performs the best on 2 data sets and FSSM on 1 data set only.

To further investigate the classification results, we conduct paired t-tests at 95% significance level and summarize the win/tie/lose counts of ISD versus other methods in Table 2.

It can be observed from Table 2 that ISD-L1 significantly outperforms EUCLID, DNE, LMNN, FSM and FSSM for 11, 11, 6, 11 and 12 times, respectively, while ISD-L2 significantly outperforms them for 9, 9, 6, 10, 11 times, respectively. ISD-L1 rarely loses to EUCLID and FSM, and only loses once against DNE, LMNN and FSSM. Similar phenomenon can be observed for ISD-L2, where ISD-L2 rarely loses to FSM, and loses only once to DNE and FSSM, twice against LMNN, and three times against EUCLID. The time cost of ISD-L1 is almost as same as LMNN and much faster than FSM/FSSM. ISD-L2 runs even faster; it is about 3.6 times faster than ISD-L1. Hence, both ISD-L1 and ISD-L2 are fairly efficient.

Table 2. The win/tie/loss counts of ISD vs. other methods, after paired t-tests at 95% significance level.

	ISD-L1	ISD-L2
EUCLID	11/5/0	9/4/3
DNE	11/4/1	9/6/1
LMNN	6/8/1	6/7/2
FSM	11/3/0	10/4/0
FSSM	12/2/1	11/3/1

As mentioned in Section 3, the learned instance specific distances can be used for updating the provided graph by reconstructing the weight matrix $\mathbf{G}$. To investigate whether such update is beneficial, extra experiments are conducted. Figure 1 plots the average error rates of the compared methods against the number of updates of the graph. In Figure 1, graph is initialized by a given one, i.e., the Gaussian fields kernel with Euclidean distance, and updated after instance specific distances obtained for each round. It is observed from Figure 1 that the error rates of ISD-L1 are reduced on 12 data sets except on *autos, diabetes, heart-statlog* and *spectf*, as the number of update increases. On *autos*, the error vibrates while the graph keeps on updating itself; this is caused by the small T value used in the experiments. By increasing T to 10, the error of ISD-L1 decreases monotonously. On *diabetes*, *heart-statlog* and *spectf*, the error increases either at the very beginning or after a few updates. Such a degradation of performance might be caused by *overfitting*, where noise is introduced in the first round and reinforced in the succeeded rounds. The performance of ISD-L2 is different. The error of ISD-L2 decreases

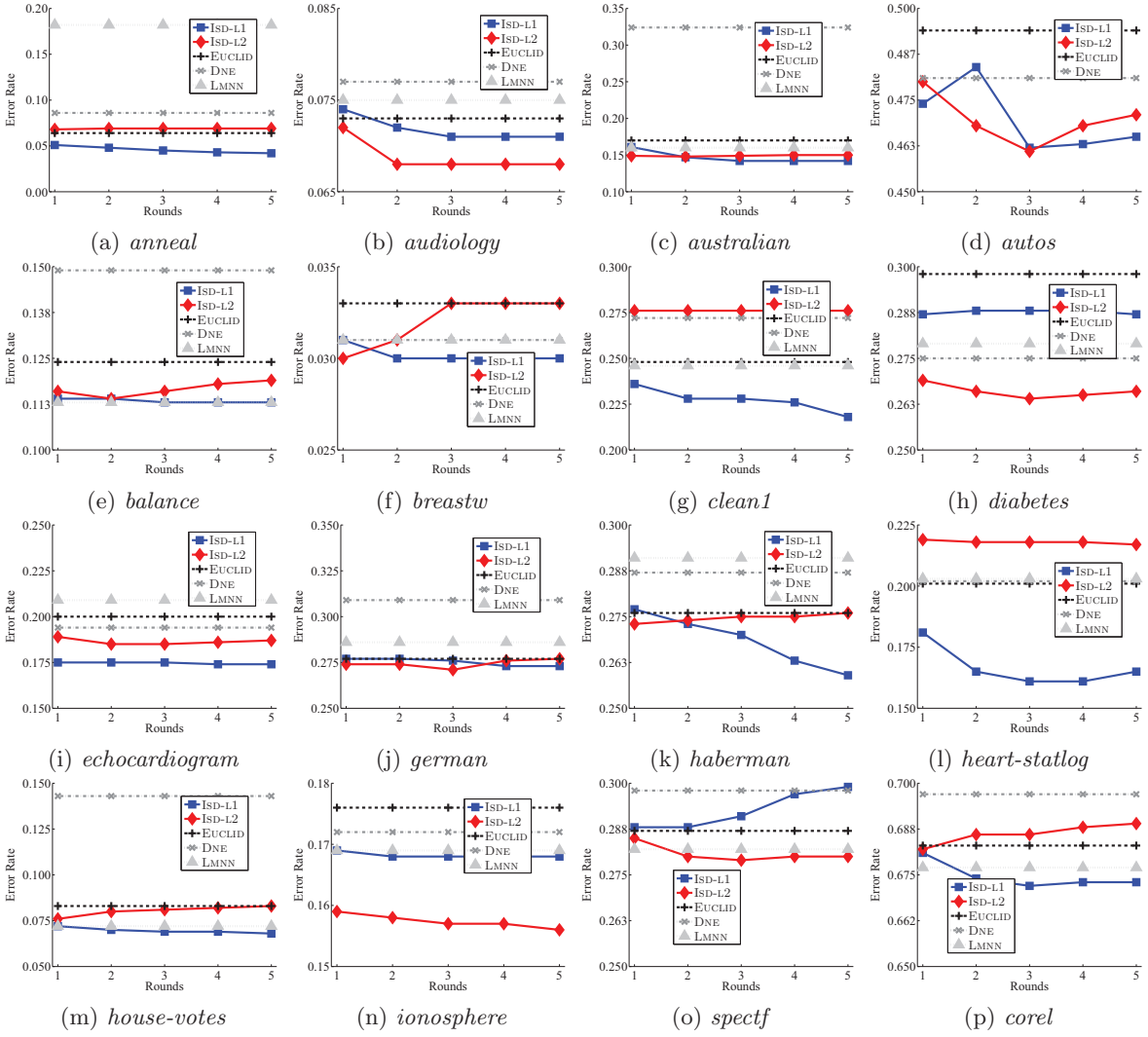

Figure 1. Influence of the iterative rounds

monotonously only on *audiology* and *ionosphere*. The degradation of performance of ISD-L2 on the other data sets suggests that ISD-L2 is more likely to overfit than ISD-L1. This is not strange for L2-Loss is more sensitive to noise so that ISD-L2 overfits more easily. Therefore, it would be better not to update the graph in ISD-L2, while updating the graph iteratively might lead to a better performance for ISD-L1 given that the maximum number of iterations for alternating descent is large enough.

Furthermore, we conduct experiments to study the influence of the amount of labeled data. Here, each data set is randomly partitioned into ten parts equally. We use 1/3/5/7/9 parts, respectively, as the labeled training data and the other parts as unlabeled test set. Experiments are repeated for 30 runs with random data partitions. Due to the page limit, we only plot the results on two data sets in Figure 2. It is observed that the advantage of ISD-L1/L2 over other methods is more obvious when the amount of labeled data are small, and ISD is less sensitive to the influence of the amount of labeled data.

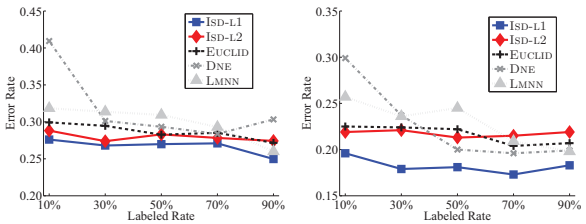

Figure 2. Influence of the amount of labeled data

5. Conclusion

Instance specific distance is desirable in many real applications, however, there is no complete framework for this purpose since existing methods can only deal with labeled examples. In this paper, we propose the ISD method, which is able to learn instance specific distances for labeled examples as well as unlabeled instances. Experiments show that ISD is superior to many state-of-the-art techniques.

The key of ISD is *metric propagation*. Although there were many studies on label propagation, to the best of our knowledge, this is the first attempt to propagating and adapting metrics from labeled examples to unlabeled instances. We accomplish the task in a convex optimization framework and attain effective and efficient solutions. It is evident that the idea of metric propagation can be applied to many other scenarios, and other kinds of metric propagation methods can be developed in the future.

Our study shows that given an initial graph, updating it gradually with the refined distances will lead to an improved performance. An interesting future issue is to study how to construct a good initial graph to enable a more effective and efficient metric propagation. It is also interesting to explore that, given a graph constructed from a data set, in addition to label propagation and metric propagation, whether other properties of instances can be propagated on the graph.

Acknowledgements

This work was supported by the NSFC (60635030, 60721002), 863 Program (2007AA01Z169), JiangsuSF (BK2008018) and Jiangsu 333 Program.

References

Belkin, M., Niyogi, P., & Sindhwani, V. (2006). Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *J. Mach. Learn. Res.*, *7*, 2399–2434.

Blake, C., Keogh, E., & Merz, C. J. (1998). UCI repository of machine learning databases. [http://www.ics.uci.edu/∼mlearn/MLRepository.html].

Frome, A., Singer, Y., & Malik, J. (2006). Image retrieval and classification using local distance functions. In *Adv. Neural. Inf. Process. Syst. 19*, 417–424.

Frome, A., Singer, Y., Sha, F., & Malik, J. (2007). Learning globally-consistent local distance functions for shape-based image retrieval and classification. *Proc. 11th Intl. Conf. Comp. Vision* (pp. 1–8).

Goldberger, J., Roweis, S., Hinton, G., & Salakhutdinov, R. (2005). Neighbourhood components analysis. In *Adv. Neural Inf. Process. Syst. 19*, 513–520.

Kwok, J., & Tsang, I. (2003). Learning with idealized kernels. *Proc. 20th Intl. Conf. Mach. Learn.* (pp. 400–407).

Li, Z., Liu, J., & Tang, X. (2008). Pairwise constraint propagation by semidefinite programming for semi-supervised classification. *Proc. 25th Intl. Conf. Mach. Learn.* (pp. 576–583).

Weinberger, K. Q., Blitzer, J., & Saul, L. K. (2005). Distance metric learning for large margin nearest neighbor classification. In *Adv. Neural Inf. Process. Syst. 17*, 1473–1480.

Xing, E. P., Ng, A. Y., Jordan, M. I., & Russell, S. (2002). Distance metric learning with application to clustering with side-information. In *Adv. Neural Inf. Process. Syst. 14*, 505–512.

Yang, L. (2006). Distance metric learning: A comprehensive survey. [http://www.cse.msu.edu/∼yangliu1/frame_survey_v2.pdf].

Zhang, R., Zhang, Z., Li, M., Ma, W.-Y., & Zhang, H. J. (2005). A probabilistic semantic model for image annotation and multi-modal image retrieval. *Proc. 10th Intl. Conf. Comp. Vision* (pp. 846–851).

Zhang, W., Xue, X., Sun, Z., Guo, Y.-F., & Lu, H. (2007). Optimal dimensionality of metric space for classification. *Proc. 24th Intl. Conf. Mach. Learn.* (pp. 1135–1142).

Zhou, D., Bousquet, O., Lal, T. N., Weston, J., & Schölkopf, B. (2003). Learning with local and global consistency. In *Adv. Neural Inf. Process. Syst. 17*, 321–328.

Zhou, Z.-H., & Dai, H.-B. (2006). Query-sensitive similarity measure for content-based image retrieval. *Proc. 6th Intl. Conf. Data Min.* (pp. 1211–1215).

Zhou, Z.-H., & Yang, Y. (2005). Ensembling local learners through multimodal perturbation. *IEEE Trans. Syst., Man and Cybern. - B*, *35*, 725–735.

Zhu, X., Ghahramani, Z., & Lafferty, J. (2003). Semi-supervised learning using Gaussian fields and harmonic functions. *Proc. 20th Intl. Conf. Mach. Learn.* (pp. 912–919).