A

Pairwised Specific Distance Learning from Physical Linkages

JUHUA HU and DE-CHUAN ZHAN, Nanjing University
XINTAO WU, University of Arkansas
YUAN JIANG and ZHI-HUA ZHOU, Nanjing University

In real tasks, it is usually the case that a better classification performance can be obtained when a good distance metric is used; therefore, distance metric learning has attracted significant attention in the past few years. Typical studies of distance metric learning concern about how to construct an appropriate distance metric that is able to separate training data points from different classes or satisfy a set of constraints (e.g., must-links and/or cannot-links). It is noteworthy that this task becomes challenging when there are only limited labeled training data points and no constraints are given explicitly. Moreover, most existing approaches aim to construct a global distance metric that is applicable to all data points. However, different data points may have different properties and may require different distance metrics. We notice that data points in real tasks are often connected by physical links (e.g., people are linked with each other in social networks; personal webpages are often connected to other webpages including non-personal webpages), but these link information has not been exploited in distance metric learning. In this paper, we develop the pairwised specific distance (PSD) approach that exploits the structure of physical linkages and in particular captures the key observations that non-metric and clique linkages imply the appearance of different or unique semantics, respectively. It is noteworthy that rather than generating a global distance, PSD will generate different distances for different pairs of data points; this property is desired in applications involving complicated data semantics. We mainly present PSD for multi-class learning and further extend it for multi-label learning. Experimental results validate the effectiveness of PSD, especially in the scenarios where there are very limited labeled training data points and no explicit constraints are given.

Categories and Subject Descriptors: I.2.6 [**Artificial Intelligence**]: Learning; H.2.8 [**Database Management**]: Database Applications—*Data mining*

General Terms: Algorithms; Design; Experimentation

Additional Key Words and Phrases: Distance metric learning, physical linkages, non-metric linkage, unlabeled data, multi-class learning, multi-label learning

ACM Reference Format:
Juhua Hu, De-Chuan Zhan, Xintao Wu, Yuan Jiang, and Zhi-Hua Zhou, 2014. Pairwised Specific Distance Learning from Physical Linkages. ACM Trans. Knowl. Discov. Data. V, N, Article A (January 2014), 27 pages.
DOI = 10.1145/2668964 http://dx.doi.org/10.1145/2668964

1. INTRODUCTION

In real tasks, it is usually the case that a good classification performance can be obtained when the distances between instances are appropriately estimated. For exam-

J. Hu and Y. Jiang was supported by the NSFC (61273301), D.-C. Zhan was supported by the NSFC (61105043), X. Wu was supported in part by NIH (1R01GM103309) and NSF (CCF-1047621), and Z.-H. Zhou was supported by the NSFC (61333014).
Authors' addresses: J. Hu and D.-C. Zhan and Y. Jiang and Z.-H. Zhou, National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China; email:{hujh, zhandc, jiangy, zhouzh}@lamda.nju.edu.cn; X. Wu, Computer Science and Computer Engineering Department, University of Arkansas, Fayetteville, AR 72701, USA; email: xintaowu@uark.edu.

ACM 1556-4681/2014/01-A $15.00
http://dx.doi.org/10.1145/2668964

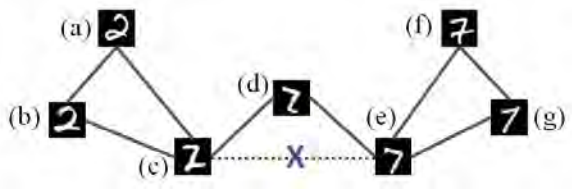


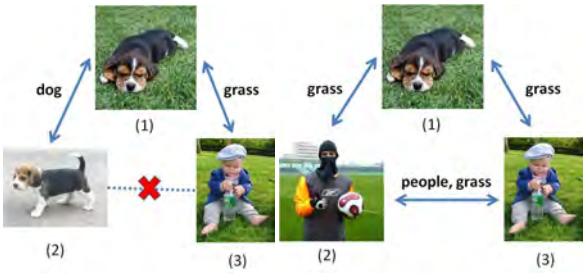
Fig. 1. An example of multi-class problem with linkages: (a)(b)(c) and (e)(f)(g) form two cliques, whereas (c)(d)(e) form a non-metric linkage (Images are from *USPS* dataset[1]).

ple, in content-based image retrieval (CBIR), the retrieval quality is highly dependent on the similarities or distances measured between images. Besides, some popular distance based classifiers such as the k-nearest-neighbor (kNN), usually conduct classification based on the calculated distances between the given test instance and all labeled instances. The instance relationship is usually constructed based on the data attributes; however, it is important to note that such relationships may be insufficient for classification in many applications. For example, pattern (d) from USPS data set[1] in Fig. 1 could not be easily distinguished because it is almost equally close to both the class '2' and class '7'. Moreover, the relationship between patterns (c), (d), and (e) shown in Fig. 1 is non-metric (i.e., (c) is similar to (d), (d) is similar to (e), but (c) is not similar to (e)); as indicated in the study [Zhang and Zhou 2009], such non-metric properties may lead to inappropriate conclusions. Distance metric learning [Yang and Jin 2006], which attempts to learn an appropriate distance metric to reflect the underlying class relationship between instances, has attracted significant attention during the past few years, and many studies [Weinberger et al. 2005; Frome et al. 2007b; Xiang et al. 2008; Tan et al. 2009; Zhan et al. 2009] have shown that appropriately learned distance metrics can significantly improve classification performance compared to the classic Euclidean distance.

Current distance metric learning approaches [Yang and Jin 2006; Frome et al. 2007b; Kumar and Kummamuru 2007; Yeung and Chang 2007; Xiang et al. 2008; Zhan et al. 2009] usually aim to construct a distance metric guided by some side information such as pairwise constraints (e.g., *must-links* and/or *cannot-links*) specified by users or induced from labeled data. Such information has been successfully exploited either globally [Xing et al. 2003] or locally [Weinberger et al. 2005]. As the advances of data collection and storage technologies, it has become much easier to collect a huge amount of data in many real tasks. However, gathering data label information is not only time-consuming but also expensive because it often requires human efforts and expertise. For example, in computer-aided medical diagnosis, a large number of X-ray images can be obtained from routine examination, yet it is difficult to request physicians to mark all focuses in all images. In other words, although there are a lot of instances, the amount of labeled data is usually limited. Therefore, side information induced from the limited labeled data is often insufficient for learning a good distance metric.

Many real data involve physical link information, e.g., people linked by friendship in social networks, webpages linked by hyperlinks, publications connected by citations. These physical linkages can be easily obtained without many human efforts, and the linkage structure may provide a new source of information. For example, one data mining paper usually cites other publications within the data mining area. The citations, i.e., the physical linkages of papers, provide a potential hint that two connected papers could be about the same topic. However, information contained in such physical linkages were totally ignored in previous distance metric learning studies.

[1] `http://www.cs.nyu.edu/~roweis/data.html`



(a) non-metric (b) clique

Fig. 2. Illustration of non-metric and clique linkages and their corresponding latent semantic meanings (Images are from the 1st version of *MSRA-MM* [Meng et al. 2009] dataset).

In this paper, physical linkages are exploited in distance metric learning, when there are limited labeled data and no additional constraints are explicitly given. It is noteworthy that the exploitation of physical linkages is non-trivial, and improper understanding will seriously mislead the learning process. This is because that the existence of different physical linkages may owe to different reasons. For an instance, image (1) in Fig. 2(a) is linked to image (2) because of the content "dog", whereas it is linked to image (3) because of "grass". The different subsets of meanings possessed by the linkages may lead to *non-metric linkages*, e.g., image (1) in Fig. 2(a) is related to both (2) and (3), but there is no linkage between (2) and (3). Such non-metric property seriously challenges traditional distance metric learning approaches [Wu et al. 2005; Yang and Jin 2006; Frome et al. 2007b; Tan et al. 2006; 2009; Zhan et al. 2009].

In fact, the appearance of a non-metric linkage within an open triplet delivers a strong implication that different semantics are possessed, e.g., "dog" and "grass" in Fig. 2(a), whereas linkages within a clique may pass same subset of semantics, e.g., "grass" in Fig. 2(b). Based on these observations, we propose a new distance metric named pairwised specific distance (PSD), aiming to exploit the latent semantics passed through different linkages.

PSD is proposed mainly under the scenario of multi-class learning. We then extend it to the multi-label learning scenario, denoted as PSD_{mc} and PSD_{ml}, respectively. For each scenario, we formulate the distance learning problem into an optimization framework. To effectively solve the optimization problem, the PSD approach is conducted in two stages: 1) learning the pairwised distance metrics for labeled data; 2) propagating the learned metrics to unlabeled data through non-metric linkages or cliques. To the best of our knowledge, this is the first study that exploits the information hidden in physical linkages for distance metric learning. Experiments on a board range of tasks validate the effectiveness of PSD.

The rest of this paper is organized as follows. Section 2 reviews some related work. Section 3 presents our proposed PSD learning approach. Section 4 shows our empirical studies, which is followed by the conclusions in Section 5.

2. RELATED WORK

Distance metric learning [Yang and Jin 2006] attempts to learn a good distance metric that reflects the underlying class relationship between instances. Traditional distance metric learning is usually guided by some constraints (or called "side information") given by users or induced from the labeled training data. Most previous distance metric learning studies focused on generating a uniform distance function for all instances by exploiting such information globally [Xing et al. 2003; Kwok and Tsang 2003] or

locally [Goldberger et al. 2005; Weinberger et al. 2005]. A recent study [Jin et al. 2009] even tried to learn a distance metric from multi-instance multi-label data.

It is noteworthy that different instances may hold different properties, and different instance pairs may have different semantic relationships that are salient in different feature subsets. Several studies [Frome et al. 2007a; Frome et al. 2007b; Zhan et al. 2009] tried to learn different distance functions for different instances. Frome *et al.* [2007a] constructed distance functions $\{D_i(\mathbf{x}_j)\}$ for each concerned labeled instance $\mathbf{x}_i$ to any other instance $\mathbf{x}_j$. Such distance functions are then optimized separately under the constraints that the concerned instance has larger distances from other instances with different labels than that from instances with the same label. Later, they [Frome et al. 2007b] extended the method by specifying some "inversed" constraints, i.e., the distance from any other instance to the concerned instance with the same label should be smaller than that from the instances with a different label. Given a test instance or a large number of unlabeled data, however, it is difficult to get their instance specific distances by using these two methods. This is because they can only generate instance specific distances for labeled instances and the instance specific distance metric for unlabeled data is left untouched. Zhan *et al.* [2009] addressed this issue by proposing the ISD (Instance Specific Distance) method in the transductive setting. The key of ISD is *metric propagation*, which propagates and adapts metrics learned for individual labeled instances to individual unlabeled instances. Thus, ISD can learn instance specific distances for labeled as well as unlabeled instances. ISD performs the metric propagation on the whole instance space, by assuming that there is a unique explanation to the affinity of instance pairs. In this paper, we consider a more challenging but practical situation, where different instance pairs may be related due to different semantics, and thus the propagations through different instance pairs have to be considered separately.

Label propagation has been widely used in graph-based semi-supervised learning (GSSL) approaches [Zhu et al. 2003; Zhou et al. 2004; Belkin et al. 2006; Wang et al. 2009]. Given a dataset, a graph $G = \{V, E\}$ can be derived, where V contains all labeled and unlabeled data points and edges in E indicate the similarities between vertices. The labels are then propagated from labeled instances to unlabeled ones over the graph. Besides labels, other properties can also be propagated across the given graph, e.g., Li *et al.* [2008] propagated pairwise constraints, Zhan *et al.* [2009] propagated the instance specific distance, and recently Kong *et al.* [2013] studied the problem of transductive multi-label learning and proposed label set propagation. In this paper, we extend the metric propagation technique to propagate semantics shared by labeled instance pairs to unlabeled ones.

GSSL and link-based classification are two typical classification models that have used the pairwise relationship between instances to facilitate the classification. GSSL is a direct way to consider the pairwise relations over the graph. Many approaches in this category have been developed, e.g., Blum and Chawla [2001] tried to use graph mincuts to separate instances from different classes; Zhu and Ghahramani [2002] used the label propagation approach; Kang *et al.* [2006] presented the correlated label propagation (CLP) approach for multi-label problems. Particularly, Zhang and Zhou [2009] proposed the non-metric label propagation (NMLP) approach, which is the first study of label propagation on graphs induced from *non-metric distances*. They converted non-metric distances into two metrics by applying spectrum transformation, so as to perform a joint label propagation on those two derived graphs.

Link-based classification [Sen and Getoor 2007] is a popular technique for mining knowledge from physical linkages. It mainly aims to classify samples using the relations or links present among them, by assuming that the labels of related objects tend to be correlated. It has received considerable attention recently and a number of ap-

proximate collective classification algorithms (ACCA) in this area have been proposed, e.g., approaches based on iterative classification (ICA) [Neville and Jensen 2000], Gibbs sampling (GS) [McEliece et al. 2007], loopy belief propagation (LBP) [Namata et al. 2009], and mean-field relaxation labeling (MF) [Namata et al. 2009]. All these techniques can be viewed as message passing algorithms that proceed in rounds, where each round consists of a set of messages being passed. In particular, each node refines its classification from label messages of its neighbors in every round in ICA, whereas messages in MF are probability distributions over class labels. These studies, however, did not consider the nature of non-metric or specifically different semantics propagated through linkages. Moreover, most pieces of these work consider only the supervised classification scenario and they treat training and testing as two separate steps with two disjoint graphs, whereas in this paper, we consider the transductive setting with only one linkage graph, as well as limited data points are labeled within the graph.

3. THE PROPOSED APPROACH

In this section, we try to exploit the different semantics possessed by different physical linkages, under the scenario that there are only limited labeled data whereas no additional constraints are explicitly given. To obtain the semantic meanings passed between unlabeled data points, we present the PSD approach to propagate the pairwised distance metrics learned from pairs of labeled data to that of unlabeled data. These learned metrics can be further used to facilitate classification.

3.1. Notations

Table I summarizes all notations that are used in this section. Concretely, we restrict our discussion in the transductive configuration and focus on the multi-class learning problem at first. Suppose there are ℓ labeled instances $\mathcal{T} = \{\mathbf{x}_i, y_i\}_{i=1}^{\ell}$ and u unlabeled instances $\mathcal{U} = \{\mathbf{x}_i\}_{i=\ell+1}^{\ell+u}$, where $\mathbf{x}_i \in \mathcal{R}^d$ and $y_i \in \mathcal{Z}$. The total number of instances is $n = \ell + u$. The available linkage graph for the whole data set is denoted as $\mathcal{L}$. This transductive multi-class classification task is to assign class labels to all those unlabeled instances.

Instead of using pairwised constraints provided by the user or induced from the labeled data as in traditional distance metric learning approaches, we extract the constraint information from the physical linkages in this work. For simplicity, if there is a physical linkage between $\mathbf{x}_i$ and $\mathbf{x}_j$, we denote it as $ij \in \mathcal{L}$. It is obvious that a single linkage $ij \in \mathcal{L}$ captures the relationship between two instances, i.e., a 2-order linkage information captured by a single linkage. In order to capture and exploit different semantic meanings for different linkages, however, higher-order linkage information is required. In this paper, we mainly consider *clique* with three points, and it is sufficient enough for higher order cliques since all 3-order cliques within a higher order clique will be considered. We denote each 3-order clique as *Clique(ijm)*, where $ij \in \mathcal{L}$, $im \in \mathcal{L}$, and $jm \in \mathcal{L}$. Similarly, *NonM(ijk)* denotes the open triplet formed by three data points $\mathbf{x}_i$, $\mathbf{x}_j$, and $\mathbf{x}_k$, where $ij \in \mathcal{L}$, $ik \in \mathcal{L}$, but $jk \notin \mathcal{L}$. For example, in Fig. 1 we may observe *Clique(abc)* and *NonM(dce)*. We denote by $ij \in S$ the condition that $\mathbf{x}_i$ and $\mathbf{x}_j$ are with the same class label, and $ij \in D$ otherwise. The distance between two instances $\mathbf{x}_i$ and $\mathbf{x}_j$ is usually calculated by the Euclidean distance based on their attributes as

$$d_{ij} = ((\mathbf{x}_i - \mathbf{x}_j)^\top (\mathbf{x}_i - \mathbf{x}_j))^{\frac{1}{2}} \tag{1}$$

In this paper, we assign a distance metric $\mathbf{w}_{ij} \in \mathcal{R}^d$ for each physically linked pair $ij \in \mathcal{L}$, and the pairwised specific distance (PSD) between them is defined as

$$d'_{ij} = (\mathbf{w}_{ij}^\top \delta_{\mathbf{x}_i,\mathbf{x}_j})^{\frac{1}{2}} \tag{2}$$

Table I. Notation Summarization

ℓ	The number of labeled data
u	The number of unlabeled data
d	The feature dimension
$\mathbf{x}_i \in \mathcal{R}^d$	The i-th instance
$y_i \in \mathcal{Z}$	The label of the i-th instance
$\mathcal{T} = \{\mathbf{x}_i, y_i\}_{i=1}^{\ell}$	The labeled training data
$\mathcal{U} = \{\mathbf{x}_i\}_{i=\ell+1}^{\ell+u}$	The unlabeled training data
$n = \ell + u$	The total number of training data
$\mathcal{L} \in \{0,1\}^{n \times n}$	The physical linkage graph
$ij \in \mathcal{L}$	There is a link between data points $\mathbf{x}_i$ and $\mathbf{x}_j$
Clique(ijm)	A 3-order clique formed by $ij \in \mathcal{L}$, $im \in \mathcal{L}$, and $jm \in \mathcal{L}$
NonM(ijk)	A non-metric open triplet formed by $ij \in \mathcal{L}$, $ik \in \mathcal{L}$, but $jk \notin \mathcal{L}$
$ij \in S$	$\mathbf{x}_i$ and $\mathbf{x}_j$ are with the same class or share some labels in multi-label problems
$ij \in D$	$\mathbf{x}_i$ and $\mathbf{x}_j$ are with different classes
d_{ij}	The Euclidean distance between $\mathbf{x}_i$ and $\mathbf{x}_j$
$\mathbf{w}_{ij} \in \mathcal{R}^d$	The distance metric for $\mathbf{x}_i$ and $\mathbf{x}_j$
d'_{ij}	The pairwised specific distance (PSD) between $\mathbf{x}_i$ and $\mathbf{x}_j$
s_{ij}	The similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$ based on the Euclidean distance d_{ij}
$\mathbf{w}_{ijp}$	The p-th entity of pairwised distance metric $\mathbf{w}_{ij}$
τ	The tradeoff parameter for non-metric linkages
η	The tradeoff parameter for cliques
$i \in \mathcal{T}$	The instance $\mathbf{x}_i$ is labeled
$i \in \mathcal{U}$	The instance $\mathbf{x}_i$ is unlabeled
c_{ij}	$c_{ij} = 1$ if $ij \in S$, otherwise, -1
T	The maximum iteration limit of Alternating Optimization algorithm
t	The t-th iteration
δ	The termination threshold for AO algorithm
$D \in [0,1]^{n \times n}$	The distances calculated for the whole data
c	The total number of labels of the multi-label problem
$\mathbf{y}_i \in \{0,1\}^c$	The label vector for the i-th mulit-label instance
ϵ_{ij}	The number of labels shared by $\mathbf{x}_i$ and $\mathbf{x}_j$ in multi-label problem
N	The number of linkages between labeled data
$\lvert\mathcal{L}\rvert$	The number of linkages for the whole data

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. It can be easily verified that when all elements of $\mathbf{w}_{ij}$ are set to 1, PSD is equivalent to the Euclidean distance. Following the study [Zhu et al. 2003], the similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$ is defined as

$$s_{ij} = \exp\left(-d_{ij}^2/\sigma^2\right) \tag{3}$$

where $\sigma = \theta \cdot \bar{d}$, $\bar{d}$ is the average distance among the whole data set, and θ is a parameter set to 1 in this paper.

3.2. The Formulation

PSD considers the distances between different instance pairs separately, owing to different semantic meanings passed through different physical linkages. As aforementioned, we introduce a pairwised distance metric $\mathbf{w}_{ij} \in \mathcal{R}^d$ for each linkage, i.e., each pair of physically associated points $ij \in \mathcal{L}$. Each element of $\mathbf{w}_{ij}$ corresponds to one feature. The larger the value of the p-th entity of $\mathbf{w}_{ij}$, i.e. $\mathbf{w}_{ijp}$, the more important the p-th feature for the linkage. Therefore, features with relatively larger weights in $\mathbf{w}_{ij}$, to some extent, indicate the latent semantic meanings passed through the linkage. Note that the physical linkages only exist between partial pairs of instances (i.e. non-complete graph), whereas distances between all instance pairs should be measured

during the classification. Therefore, we set the pairwised distance metric $\mathbf{w}_{ij} = \mathbf{1}$ for non physically linked instance pairs, which equals to the Euclidean distance in fact.

3.2.1. Supervision information. In this section, we focus on the PSD learning for multi-class problems, where one instance belongs to one single class, denoted as PSD$_{mc}$. A physical linkage usually occurs when two instances are related; however, the reason for them to be linked may vary for different instance pairs. Taking webpages as an example, a "faculty" webpage can be linked to a "course" webpage because the faculty teaches this course, and at the same time, a "project" webpage distributed from this course makes a linkage between the "course" and "project" webpages; these three webpages, however, are from totally different classes, i.e., "faculty", "course", and "project". It is evident that linkages are formed by various reasons, which may even bring about linkages for two instances that are from different classes, and these linkages will mislead the classification task. To address this problem, we use the pairwised distance metric $\mathbf{w}_{ij}$ to shrink instances within the same class closer, whereas separating instances from different classes far away, following the idea of supervised global distance metric learning [Yang and Jin 2006]. In common distance metric learning, e.g. in the study [Xing et al. 2003], the distance between instances from different classes is constrained to be larger than 1. In this paper, we simply constrain the distance between instances from the same class to be smaller than their Euclidean distance, and the distance between instances from different classes larger than their Euclidean distance by using the following constraints.

$$d'_{ij} < d_{ij}, ij \in S$$
$$d'_{ij} > d_{ij}, ij \in D$$

This setting has been confirmed to be able to improve classification performance in the study [Geng et al. 2005]. The information that if two instances are from the same class can then be incorporated into the pairwised distance metrics for pairs of labeled data through these two constraints, so as to be propagated to the pairs of unlabeled data.

3.2.2. Non-metric linkage. Similar to the non-metric example shown in Fig. 1, naturally existing physical linkages or linkages indicated by users in most real tasks may be based on mono-specific semantic meanings, which will lead to *non-metric linkages*. For an instance, one database paper cites another database publication because they tackle the same data management problem, while the paper also cites a machine learning publication because of the adoption of the learning technique developed in that machine learning paper. However, there is no citation between these two cited ones, which represents a non-metric linkage. In other words, a non-metric linkage implies that the two linkages within it are carrying totally different semantic meanings. Formally, the two linkages $ij \in \mathcal{L}$ and $ik \in \mathcal{L}$ within the non-metric linkage *NonM(ijk)* should possess different semantic meanings, and thus their pairwised distance metrics $\mathbf{w}_{ij}$ and $\mathbf{w}_{ik}$ should be different. We use $\mathbf{w}_{ik}^\top \mathbf{w}_{ij}$ to indicate the difference between two metrics. It is obvious that the smaller the product is, the larger the difference. Intuitively, the larger the difference between instances $\mathbf{x}_j$ and $\mathbf{x}_k$, the less the chance that $ij \in \mathcal{L}$ and $ik \in \mathcal{L}$ pass some similar semantic meanings. Therefore, to capture the extent of difference between $\mathbf{w}_{ij}$ and $\mathbf{w}_{ik}$ for each non-metric linkage *NonM(ijk)*, it is natural to minimize

$$d_{jk} \mathbf{w}_{ik}^\top \mathbf{w}_{ij} \tag{4}$$

where d_{jk}, the Euclidean distance between $\mathbf{x}_j$ and $\mathbf{x}_k$, reflects their attribute difference.

3.2.3. Clique. Consider the two cliques *Clique(abc)* and *Clique(efg)* as examples in Fig. 1, it is obvious that three linkages within each clique share a subset of semantic

meanings. It implies that the pairwised distance metrics for each two linkages within this clique could be similar on a subset of features, which in other words means their pairwised distance metrics should be similar to some extent. Formally in *Clique(ijm)*, the smaller the value of $\|\mathbf{w}_{im} - \mathbf{w}_{jm}\|_2^2$ is kept, the more similar the pairwised distane metrics for the two linkages $im \in \mathcal{L}$ and $jm \in \mathcal{L}$ is constrained. Besides in intuition, the larger the similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$ on their attributes, the more semantic meanings should be shared by $im \in \mathcal{L}$ and $jm \in \mathcal{L}$, and thus, the larger the extent that $\mathbf{w}_{im}$ and $\mathbf{w}_{jm}$ are similar. Thereafter, for each *Clique(ijm)*, we minimize Eq. 5 to capture the similarity level for each pair of distance metrics.

$$s_{ij}\|\mathbf{w}_{im} - \mathbf{w}_{jm}\|_2^2 + s_{jm}\|\mathbf{w}_{ij} - \mathbf{w}_{im}\|_2^2 + s_{im}\|\mathbf{w}_{ij} - \mathbf{w}_{jm}\|_2^2 \tag{5}$$

where s_{ij}, s_{im} and s_{jm} reflect the attribute similarities between each two instances.

3.2.4. Optimization framework. In order to infer the pairwised distance metric for each physically linked instance pair, we formulate the problem into the following optimization framework

$$\arg\min_{\mathbf{w}} \quad \sum_{ij\in\mathcal{L}} \|\mathbf{w}_{ij}\|_2^2 + \tau\Omega + \eta\Xi \tag{6}$$

$$s.t. \quad \sum_{p} \mathbf{w}_{ijp} = d \tag{7}$$

$$d'_{ij} < d_{ij}, \text{ if } ij \in S \tag{8}$$

$$d'_{ij} > d_{ij}, \text{ if } ij \in D \tag{9}$$

where τ and η are two tradeoff parameters. In Eq. 7 we fairly constrain the sum of elements for each metric $\mathbf{w}_{ij}$ to be the feature dimensionality d as the Euclidean distance. The supervision information from limited labeled data are used in the Eqs. 8 and 9. The regularization term Ω takes the property of non-metric linkage into account as

$$\Omega : \sum_{k,NonM(ijk)} d_{jk}\mathbf{w}_{ik}^{\top}\mathbf{w}_{ij} \tag{10}$$

where all non-metric linkages containing $ij \in \mathcal{L}$ are included. Ξ is another regularization term used to consider the hidden semantic meanings shared within cliques as

$$\Xi : \sum_{m,Clique(ijm)} s_{ij}\|\mathbf{w}_{im} - \mathbf{w}_{jm}\|_2^2 + s_{jm}\|\mathbf{w}_{ij} - \mathbf{w}_{im}\|_2^2 + s_{im}\|\mathbf{w}_{ij} - \mathbf{w}_{jm}\|_2^2 \tag{11}$$

which considers all cliques containing $ij \in \mathcal{L}$. It should be noted that since d_{jk} in Ω estimated by Eq. 1 can be naturally much larger than s_{ij} estimated by Eq. 3 in Ξ, we normalize the Euclidean distances used in Ω into $[0, 1]$ in our experiment.

3.3. The Solution

It is computationally expensive to solve the PSD$_{mc}$ learning problem in Eq. 6 directly because the number of physical linkages could be large. Besides, the bilinear problem induced in Eq. 10 makes the whole optimization problem non-convex. To address this problem, we decompose the optimization problem into two stages by considering the labeled data and unlabeled data separately. The main idea is inspired by label propagation [Zhu and Ghahramani 2002], where the label information given by labeled data are firstly propagated to those unlabeled data points that are directly linked to the labeled data, and then to other unlabeled data points further and further. Concretely, in the first stage, we obtain the pairwised distance metrics for all pairs of labeled data that are also physically linked. Then in the second stage, the learned metrics from

labeled pairs are propagated to those pairs containing both labeled and unlabeled data at first, and to other unlabeled pairs further and further. Unlink label propagation that propagates label information through graph edges, the pairwised distance metrics are propagated through non-metric linkages and 3-order cliques. This process is named *pairwised metric propagation* in this paper, and the details are described in the following two stages.

3.3.1. Stage one: Obtaining semantic meanings possessed by linkages between labeled data. In this first stage, we consider only the pairwised distance metrics for labeled data that are also physically linked. By subtracting all relevant parts for labeled data in Eq. 6, we have the following objective function

$$\begin{aligned} \arg\min_{\mathbf{w}} \quad & \sum_{ij\in\mathcal{L},i,j\in\mathcal{T}} \|\mathbf{w}_{ij}\|_2^2 + \tau\Omega + \eta\Xi \\ s.t. \quad & \sum_p \mathbf{w}_{ijp} = d \\ & c_{ij}d'_{ij} < c_{ij}d_{ij} \end{aligned} \tag{12}$$

where we define $c_{ij} \in \{-1, 1\}$ with $c_{ij} = 1$ indicating $ij \in S$ and $c_{ij} = -1$ indicating $ij \in D$. By replacing the two regularization terms defined in Eq. 10 and Eq. 11, the objective function in Eq. 12 can be rewritten to

$$\begin{aligned} \arg\min_{\mathbf{w}} \quad & \sum_{ij\in\mathcal{L},i,j\in\mathcal{T}} [(1+\eta\sum_m (s_{jm}+s_{im}))\|\mathbf{w}_{ij}\|_2^2 + (\tau\sum_k d_{jk}\mathbf{w}_{ik}^\top \\ & -2\eta\sum_m (s_{jm}\mathbf{w}_{im}^\top + s_{im}\mathbf{w}_{jm}^\top))\mathbf{w}_{ij} + \eta\sum_m ((s_{ij}+s_{jm})\|\mathbf{w}_{im}\|_2^2 \\ & +(s_{ij}+s_{im})\|\mathbf{w}_{jm}\|_2^2 - 2s_{ij}\mathbf{w}_{im}^\top\mathbf{w}_{jm})] \\ s.t. \quad & \sum_p \mathbf{w}_{ijp} = d \\ & c_{ij}d'_{ij} < c_{ij}d_{ij} \end{aligned} \tag{13}$$

Now let $f(\mathbf{w})$ be the objective function in Eq. 13, this problem can be solved by the well known Alternating Optimization (AO) algorithm [Bezdek and Hathaway 2003] through the following steps.

(1) Naturally partition $\mathbf{w}$ as $\mathbf{w} = (\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_N)^\top$, where each $\mathbf{w}_i$ represents a pairwised distance metric and N is the total number of linkages that belong to $\mathcal{L}$ with its two nodes labeled.
(2) Initialize $\mathbf{w}^{(0)} = (\mathbf{w}_1^{(0)}, \cdots, \mathbf{w}_N^{(0)})^\top$ where $\mathbf{w}_i^{(0)} = \mathbf{1}$, termination threshold δ, maximum iteration limit T, and iteration counter $t = 0$.
(3) For each $i = 1, \cdots, N$, compute

$$\mathbf{w}_i^{(t+1)} = \arg\min f(\mathbf{w}_1^{(t+1)}, \cdots, \mathbf{w}_{i-1}^{(t+1)}, \mathbf{w}_i, \mathbf{w}_{i+1}^{(t)}, \cdots, \mathbf{w}_N^{(t)})$$

(4) If $\|\mathbf{w}^{(t+1)} - \mathbf{w}^{(t)}\| \le \delta$ or $t \ge T$, quit; otherwise, set $t = t + 1$ and go to Step 3.

Obviously, computing $\mathbf{w}_i^{(t+1)}$ is a quadratic programming problem, and thus has the global minimizer. More importantly, we restrict not to increase the objective function value at each step, and thus local convergence of this AO algorithm can be guaranteed. In the experiment, we only set the maximum iteration limit $T = 20$ and superb performance is obtained, which empirically verifies the fast convergence rate.

We can easily observe that the space complexity of this stage is $\mathcal{O}(Nd)$. From the AO steps, we can find that in the worst case, we need to run alternating for T iterations. In each iteration, the quadratic programming should be calculated for N times. Since quadratic programming is linear in d with the best implementation. The computational cost of this stage is $\mathcal{O}(NdT)$. Therefore, the space and time complexity is highly dependent on the total number of linkages involved (i.e., linkages with both nodes labeled) and the feature dimension d. Since there are limited labeled data, N is also limited in this stage. Although it could be time-consuming for very dense graphs, it is scalable for sparse graphs, which are more often in real tasks.

3.3.2. Stage two: Inferring the latent semantic meanings possessed by linkages between unlabeled data by propagation. In this second stage, the pairwised distance metrics learned from the first stage are firstly propagated to the linkages that directly connect labeled and unlabeled data, and then to farther unlabeled data pairs. For each concerned instance pair $\mathbf{x}_i$ and $\mathbf{x}_j$, we have the following optimization problem subtracted from Eq. 6

$$\begin{aligned} \arg\min_{\mathbf{w}} \quad & \sum_{ij\in\mathcal{L}} \|\mathbf{w}_{ij}\|_2^2 + \tau\Omega + \eta\Xi \\ s.t. \quad & \sum_{p} \mathbf{w}_{ijp} = d \end{aligned} \tag{14}$$

Similar to the first stage, we can solve this optimization problem by the AO algorithm, which requires the similar space and time complexity. Since the number of linkages between unlabeled data is much larger, this stage requires longer time than the first stage. By dividing the problem into two stages, we also distribute the computational cost into two stages, which is much less than solving the whole problem in one stage. Since the number of linkages between unlabeled data is usually much larger than that of labeled data, the time complexity is dominated by the second stage, which is $\mathcal{O}((|\mathcal{L}| - N)dT)$.

Based on the learned pairwised distance metrics, the PSDs for the whole data can be calculated by Eq. 2, and Euclidean distances are computed for non physically linked pairs. Then, we can use the updated distances to facilitate the classification on unlabeled data. The pseudo-code of PSD$_{mc}$ is summarized in Algorithm 1.

3.4. Extension to Multi-Label Learning

Unlike multi-class learning where each instance belongs to one single class, in multi-label problem, one instance is associated with multiple labels, i.e. each instance is compound with multiple different semantic meanings, as those images shown in Fig. 2. Obviously, the phenomenon that different linkages possess different semantic meanings is much more significant than multi-class problems. Blindly propagating label information through existing physical linkages will seriously lead to incorrect results, because some linkages may pass only one specific label information, whereas others may possess more than one label information, e.g. Fig. 2(b). Therefore, distinguishing different semantics shared by different instance pairs is a key challenge for distance metric learning in multi-label problems.

In real-world problems such as image or video annotation [Hua and Qi 2008], it is often the case that only limited labels are annotated. In other words, not all instances are labeled, and more seriously, not all labels for one instance are given [Sun et al. 2010; Zhang et al. 2011; Wang et al. 2011; Yang et al. 2013], which is called the weak label problem. Consequently, given labels positively indicate the current instance containing the corresponding semantic. However, this instance may contain more semantics,

ALGORITHM 1: PSD_{mc}

Input: $\mathcal{T} = \{\mathbf{x}_i, y_i\}_{i=1}^{\ell}$: ℓ labeled instances;
$\mathcal{U} = \{\mathbf{x}_i\}_{i=\ell+1}^{\ell+u}$: u unlabeled instances;
$\mathcal{L}$: physical linkages for the whole data set;
T: maximum iteration limit;
δ: termination threshold.
Output: $D \in [0,1]^{n \times n}$.
Process:
Initialize $\mathbf{w}^{(0)} = \mathbf{1}$;
for $t = 1, \cdots, T$ **do**
  for $ij \in \mathcal{L} \wedge i, j \in \mathcal{T}$ **do**
    Learn $\mathbf{w}_{ij}$ by solving Eq. 13 with other $\mathbf{w}'s$ fixed;
  end
  if $\|\mathbf{w}^{(t)} - \mathbf{w}^{(t-1)}\| \leq \delta$ **then**
    Break;
  end
end
$\mathbf{w}^{(0)} = \mathbf{w}$;
for $t = 1, \cdots, T$ **do**
  for $ij \in \mathcal{L} \wedge (i \in \mathcal{U}, j \in \mathcal{T} \vee i, j \in \mathcal{U})$ **do**
    Propagate learned metrics to $\mathbf{w}_{ij}$ by solving Eq. 14 with other $\mathbf{w}'s$ fixed;
  end
  if $\|\mathbf{w}^{(t)} - \mathbf{w}^{(t-1)}\| \leq \delta$ **then**
    Break;
  end
end
Calculate PSDs by Eq. 2 into D;
Calculate Euclidean distances for remaining pairs by Eq. 1 into D;

although the label is currently not annotated. Based on the naturally existing linkages as shown in Fig. 2, our pairwised distance metric can be used to explore the semantic meanings passed through different physical linkages, so as to propagate the corresponding specific label information to its neighborhoods. Social network community partition [Wu et al. 2011] is another realistic application scenario, where people are usually associated due to different reasons, e.g., "classmates", "colleges", or "hobbies" and so on, which may form different communities. One person may belong to multiple communities because he/she plays different roles in different situations. One possible way to address this community partition problem is to figure out the semantic meanings passed through those friendships, and all people involved within each semantic can form a community.

We confine this multi-label extension within the transductive scenario, let $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{n}$ denote the training data, with $\mathbf{x}_i \in \mathcal{R}^d$, and $\mathbf{y}_i \in \{0,1\}^c$, where 1 indicates that $\mathbf{x}_i$ has the corresponding label, whereas 0 indicates that we do not know if this instance has this specific label. The physical linkages between these instances are also given in $\mathcal{L}$. The task is to label all instances completely by exploiting $\mathcal{L}$. To facilitate the labeling, we still use the pairwised distance metric $\mathbf{w}_{ij}$ with different weighting values on different features to imply the hidden semantic meanings possessed by each physical linkages $ij \in \mathcal{L}$. The properties of non-metric linkages and cliques can be included similarly. However, unlike multi-class learning, where each instance is related to only one class and thus it is easy to figure out if two labeled instances are from the same class to have Constraints 8 and 9, it is impossible to determine if two instances are with totally different labels in multi-label learning. It is because that each instance is associated

with multiple labels, but not all its labels are given in weak label problems. In other words, from those given labels, we can only determine how many labels are currently shared by an instance pair. We use the number of shared labels between instances $\mathbf{x}_i$ and $\mathbf{x}_j$ to describe the extent that they are related, denoted by ϵ_{ij}. By changing the way of using supervision information in the objective function in Eq. 6, we have the following optimization framework for weak label problems.

$$\begin{aligned} \arg\min_{\mathbf{w}} \quad & \sum_{ij\in\mathcal{L}} \|\mathbf{w}_{ij}\|_2^2 + \tau\Omega + \eta\Xi && (15) \\ s.t. \quad & \sum_p \mathbf{w}_{ijp} = d \\ & d'_{ij} < d_{ij} - \epsilon_{ij}, \text{ if } ij \in S && (16) \end{aligned}$$

where the new constraint 16 tries to shrink those instance pairs that share more labels closer, and the regularization term Ω and Ξ are defined similarly as in Eqs. 10 and 11, respectively. To distinguish with PSD$_{mc}$, this extension is named PSD$_{ml}$.

Although there is no way to separate labeled and unlabeled data explicitly for the weak label problem in Eq. 15, we in the first stage learn the pairwised distance metrics between instance pairs with label information. Then in the second stage, those learned distance metrics are propagated over the whole linkage graph (including instances with or without label information) through the non-metric linkages and cliques. Similarly, we divide the PSD$_{ml}$ learning problem into the following two stages, Eq. 17 and Eq. 18, respectively.

$$\begin{aligned} \arg\min_{\mathbf{w}} \quad & \sum_{ij\in\mathcal{L}\wedge ij\in S} \|\mathbf{w}_{ij}\|_2^2 + \tau\Omega + \eta\Xi && (17) \\ s.t. \quad & \sum_p \mathbf{w}_{ijp} = d \\ & d'_{ij} < d_{ij} - \epsilon_{ij} \end{aligned}$$

$$\begin{aligned} \arg\min_{\mathbf{w}} \quad & \sum_{ij\in\mathcal{L}} \|\mathbf{w}_{ij}\|_2^2 + \tau\Omega + \eta\Xi && (18) \\ s.t. \quad & \sum_p \mathbf{w}_{ijp} = d. \end{aligned}$$

It can be easily observed that in the second stage in Eq. 18, the number of pairwised distance metrics that we need to learn is the total number of linkages in $\mathcal{L}$. We cannot distribute the computational cost into two stage for the weak label problem. To solve this problem similarly using the AO algorithm as the multi-class scenario, the time complexity is $\mathcal{O}(|\mathcal{L}|dT)$, where $|\mathcal{L}|$ is the total number of linkages for the whole data. The pseudo-code of PSD$_{ml}$ is summarized in Algorithm 2.

4. EXPERIMENTS

To evaluate the effectiveness of PSD, we conduct the empirical study for both scenarios discussed as well, i.e., multi-class learning and multi-label learning. We focus on the multi-class scenario with detailed empirical evaluations, and conduct a small extension on the multi-label scenario to visualize semantic meanings distinguished through different linkages.

ALGORITHM 2: PSD_{ml}

Input: $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^n, \mathbf{x}_i \in \mathcal{R}^d, \mathbf{y}_i \in \{0,1\}^c$: n instances with their label vectors;
$\mathcal{L}$: physical linkages for the whole data set;
T: maximum iteration limit;
δ: termination threshold.
Output: $D \in [0,1]^{n \times n}$.
Process:
Initialize $\mathbf{w}^{(0)} = \mathbf{1}$;
for $t = 1, \cdots, T$ **do**
  for $ij \in \mathcal{L} \wedge ij \in S$ **do**
    Learn $\mathbf{w}_{ij}$ by solving Eq. 17 with other $\mathbf{w}'s$ fixed;
  end
  if $\|\mathbf{w}^{(t)} - \mathbf{w}^{(t-1)}\| \leq \delta$ **then**
    Break;
  end
end
$\mathbf{w}^{(0)} = \mathbf{w}$;
for $t = 1, \cdots, T$ **do**
  for $ij \in \mathcal{L}$ **do**
    Propagate learned metrics to $\mathbf{w}_{ij}$ by solving Eq. 18 with other $\mathbf{w}'s$ fixed;
  end
  if $\|\mathbf{w}^{(t)} - \mathbf{w}^{(t-1)}\| \leq \delta$ **then**
    Break;
  end
end
Calculate PSDs by Eq. 2 into D;
Calculate Euclidean distances for remaining pairs by Eq. 1 into D;

4.1. Multi-Class Scenario

In this section, we focus on the multi-class scenario, and thus on the evaluation of PSD_{mc}.

4.1.1. Configuration. We evaluate PSD_{mc} on four datasets with both physical linkages and label information [Sen et al. 2008], including two bibliographic datasets, i.e. *cora* and *citeseer*, with publications connected by citations, one webpage dataset *WebKB* containing webpages linked by hyperlinks from four isolated universities, i.e. *cornell*, *texas*, *washington*, and *wisconsin*, and one social network dataset *Terrorist Attacks* which is composed by different types of attacks and contains two different linkage types. One is for attacks invoked in the same location and the other is for attacks not only invoked in the same location but also organized by the same organization, denoted as *ta-loc* and *ta-loc-org*, respectively.

After removing all self-linkages and isolated nodes, the statistics of each dataset are summarized in Table II. To study the effect of the number of labeled data, we randomly sample $\{5\%, 10\%, 20\%, 40\%, 80\%\}$ of data as labeled data and the rest as unlabeled data for each dataset. Each process is repeated for 30 trials and the average prediction performance on unlabeled data is reported.

To evaluate how PSD_{mc}s can help for classification, we execute the following three GSSL approaches over the graphs induced from the new distances. They all have the parameter k, which indicates the number of nearest neighbors considered. We conduct the experiments with different k values from {1,3,5,7} to study the effect of the different k's.

— **LapSVM**: Laplacian support vector machine [Melacci and Belkin 2011].

Table II. Dataset Summarization.

Data set	#inst.	#class	#feat.	#links	#links per node			#nodes with k links		
					min	max	$mean \pm std$	$k=1$	$k=2$	$k \geq 3$
cora	2,708	7	1,433	5,278	1	164	3.90 ± 5.23	485	583	1,640
citeseer	3,264	6	3,703	4,536	1	99	2.78 ± 3.40	1,321	796	1,147
ta-loc	645	6	106	3,172	1	49	9.84 ± 12.6	151	63	431
ta-loc-org	260	5	106	571	1	15	4.39 ± 4.52	95	35	130
cornell	195	5	1,703	283	1	94	2.90 ± 6.83	88	40	67
texas	185	5	1,703	280	1	104	3.03 ± 7.79	72	53	60
washington	217	5	1,703	366	1	122	3.37 ± 8.33	58	64	95
wisconsin	262	5	1,703	459	1	122	3.50 ± 7.79	72	71	119

Table III. Effects of tradeoff parameters τ and η on prediction accuracy by applying LapSVM (The label ratio is set to 20%, $k=3$, η is from $\{1,5,10\}$, and τ is from $\{1,5,10,100\}$. For each η, the average over all τ's is reported).

η	*ta-loc*	*ta-loc-org*	*cornell*	*texas*	*washington*	*wisconsin*
1	.803±.002	.816±.000	.515±.008	.615±.002	.688±.002	.676±.013
5	.800±.004	.817±.000	.527±.000	.613±.000	.707±.012	.657±.000
10	.796±.006	.814±.004	.526±.013	.613±.003	.707±.004	.653±.004
Ave.	.800±.004	.816±.003	.523±.007	.614±.002	.701±.007	.662±.008

— **LP**: label propagation [Zhu et al. 2003].
— **NNP**: k-nearest-neighbor propagation. Each unlabeled instance will receive the label information from its k nearest neighbors, and then its class label is decided by majority voting; this process will be repeated until convergence.

In the experiment, we set the maximum iteration limit for the AO algorithm to be $T=20$ and the termination threshold as $\delta=0$ that means stopping when no metric is updated. Although the tradeoff parameters τ and η for non-metric linkages and cliques could be tuned by cross validation for large data sets *cora* and *citeseer*, cross validation is not reliable or even cannot be conducted on other small data sets, since the number of labeled data is too small. We fix both of them to be $\tau=\eta=10$ for all experiments, because in our experience they will not affect the performance too much. For example, when the label ratio is set to 20% and $k=3$, by using different settings of τ (1,5,10,100) and η (1,5,10), we apply the LapSVM over the graphs induced from the PSDs. Table III shows the average prediction accuracy and the corresponding standard deviation for all τ's when η is fixed, and the average accuracy of all settings for each data set. It can be observed from the standard deviations that they do not affect the prediction performance much in most cases.

To verify the superbness of PSD_{mc}, we also apply the three GSSL approaches over the graphs induced by the following four ways.

— **Cont**: It is constructed solely based on the data attributes by using the Euclidean distance
— $\mathcal{L}$: It directly uses the linkage graph $\mathcal{L}$ with elements 0 or 1.
— **CC**: For each instance pair $\mathbf{x}_i$ and $\mathbf{x}_j$ from the labeled training data $\mathcal{T}$, we generate a new instance by $|\mathbf{x}_i - \mathbf{x}_j|$. Each new instance has a new class label $y' \in \{0,1\}$, where 1 indicates $\mathbf{x}_i$ and $\mathbf{x}_j$ are from the same class and 0 otherwise. A soft classifier can be learned from the generated data, and be used to get the probability that two instances share the same label, which is called the common confidence in this paper.
— **CC'**: We treat linked instance pairs separately from unlinked pairs. For each set, we train a soft classifier to get the common confidences as CC. Then, we combine the common confidences from two sets together to get CC'.

Table IV. CPU time comparison (in minutes; label ratio is set to 20% for each dataset).

Data	CC	CC'	ISD	PSD
cora	N/A	N/A	349.05	**53.81**
citeseer	N/A	N/A	1914.33	**94.73**
ta-loc	532.13	214.42	7.04	**3.83**
ta-loc-org	15.94	7.16	2.44	**0.50**
cornell	70.86	31.55	13.84	**5.21**
texas	59.37	24.88	12.10	**4.67**
washington	84.33	48.62	15.15	**6.73**
wisconsin	214.27	101.90	15.57	**7.61**

In addition, the following two methods with special focuses are also compared with the LapSVM applied on the graph induced by PSD_{mc}. We only compare with LapSVM because it is both more effective and efficient than LP and NNP.

— **ISD**: Instance specific distance is proposed in [Zhan et al. 2009] that compute the distances based on each instance's specification.
— **NMLP**: Non-metric label propagation [Zhang and Zhou 2009] can do label propagation over graphs induced from non-metric distances. Considering that the physical linkages are with non-metric phenomenon, NMLP is applied directly on the linkage graph. Note that the original NMLP is for binary classification, in our experiments we extend it to multi-class problems by using the one-vs-all strategy, and choose the class with the highest confidence as the final prediction.

All experiments are performed on a machine with 12 Intel Xeon processors/cores (1.6GHz) with 12GB RAM. Since CC, CC', ISD, and PSD require additional time to learn new distances or similarity matrices, their average time cost of 30 trials are compared in Table IV when the label ratio is set to 20%. Due to the relatively larger size for datasets *cora* and *citeseer*, learning CC or CC' suffers from heavy time cost and even memory problem because $\mathcal{O}(\binom{n}{2})$ new instances are generated. Surprisingly, the time cost of learning PSD_{mc} is much lower than CC, CC' and ISD.

To study the effect of parameter k, the percentage of labeled data for each data set is firstly fixed to 20%, and k changes from set $\{1, 3, 5, 7\}$. Further to evaluate PSD_{mc} with different amount of labeled data, we conduct experiments when $k = 3$ and the percentage of labeled data varies from 5% to 80%.

4.1.2. Publication Categorization. In this section, we apply our PSD_{mc} method on the two bibliographic datasets *cora* and *citeseer*. Papers in the *cora* are from the machine learning field and each is categorized into one of seven possible topics, i.e., case based, genetic algorithms, neural networks, probabilistic methods, reinforcement learning, rule learning, and theory. The *citeseer* dataset includes papers from six categories, i.e., agents, artificial intelligence, database, human computer interaction, machine learning, and information retrieval.

For each dataset, three GSSL approaches (i.e., LapSVM, LP, NNP) are applied over three different graphs induced from Cont, $\mathcal{L}$ and PSD, since CC and CC' are not available for these two datasets. First to verify that PSD can provide more appropriate distance measures than others, no matter how many nearest neighbors are considered, the label ratio is fixed to 20% and k changes from $\{1, 3, 5, 7\}$. Table V shows the average prediction accuracy on unlabeled data of 30 trials for each setting with the best performance in bolded fonts. It can be easily observed that PSD achieves significantly better performance, no matter what kind of GSSL approach is applied or how many nearest neighbors are considered. Besides, PSD with LapSVM applied achieves significantly

Table V. Accuracy comparison on *cora* and *citeseer* with fixed label ratio 20%. The best performance and its comparable performances are bolded (statistical significance examined via pairwise t-tests at 95% confidence level).

Data	Method	Metric	$k=1$	$k=3$	$k=5$	$k=7$
cora	LapSVM	Cont	.460±.022	.447±.013	.405±.011	.382±.019
		$\mathcal{L}$	.610±.018	.596±.012	.537±.012	.502±.019
		PSD	**.783±.008**	**.808±.006**	**.808±.005**	**.808±.005**
	LP	Cont	.427±.020	.429±.037	.397±.044	.393±.044
		$\mathcal{L}$	.571±.018	.584±.035	.543±.040	.530±.040
		PSD	**.761±.012**	**.790±.010**	**.792±.010**	**.792±.010**
	NNP	Cont	.282±.045	.232±.060	.229±.080	.233±.095
		$\mathcal{L}$	.456±.035	.370±.054	.335±.078	.333±.089
		PSD	**.544±.060**	**.531±.071**	**.421±.114**	**.308±.109**
	NMLP	-	.739±.020	.759±.021	.702±.041	.620±.065
	ISD	-	.355±.050	.326±.058	.322±.075	.316±.079
	LapSVM	PSD	**.783±.008**	**.808±.006**	**.808±.005**	**.808±.005**
citeseer	LapSVM	Cont	.391±.011	.353±.006	.310±.006	.283±.006
		$\mathcal{L}$	.494±.013	.454±.044	.410±.046	.384±.049
		PSD	**.624±.011**	**.641±.006**	**.642±.006**	**.641±.006**
	LP	Cont	.358±.011	.334±.043	.303±.046	.277±.043
		$\mathcal{L}$	.482±.012	.462±.042	.421±.046	.391±.043
		PSD	**.572±.011**	**.595±.008**	**.599±.007**	**.598±.006**
	NNP	Cont	.290±.024	.227±.032	.197±.035	.174±.050
		$\mathcal{L}$	.334±.018	.246±.010	.228±.034	.177±.050
		PSD	**.439±.032**	**.266±.039**	**.233±.078**	**.191±.054**
	NMLP	-	.576±.014	.606±.015	.577±.018	.532±.022
	ISD	-	.322±.037	.307±.057	.341±.061	.354±.067
	LapSVM	PSD	**.624±.011**	**.641±.006**	**.642±.006**	**.641±.006**

higher prediction accuracy on unlabeled data compared to ISD and NMLP no matter how many nearest neighbors are considered.

The advantage of PSD, when there are different amount of labeled data, is further shown in Fig. 3 with two measures, prediction accuracy and F1 measure. It should be pointed out that we rule ISD's performance out in this respect, because ISD not only takes more than 5 hours for each trial as indicated in Table IV but also performs worse than any other method as shown in Table V. Fig. 3 shows that PSD provides better prediction performance in terms of both accuracy and F1 measure in most cases, no matter what kind of GSSL approach is applied or how many data points are labeled. Moreover, it can be easily seen that the helpfulness of PSD is more significant when there are less labeled instances.

Besides, we find that directly using physical linkages surprisingly leads to better performance than Euclidean distance on these two datasets. This suggests that physical linkages provide helpful category information, and the data attributes themselves are insufficient for classification for these two data sets. This phenomenon leads to better prediction performance of NMLP which directly utilizes the link graph, and worse prediction performance of ISD which totally relies on the data attributes. It is not difficult to understand: citations often occur between papers about similar topics, and this link information can give valuable classification information. Since publications in *cora* are all from the machine learning field and those in *citeseer* are all from computer science, there would be big attribute similarity between publications within the same field although they are from different topics. This makes the data attributes insufficient for distance measure between instances. In this case, PSD is further helpful by

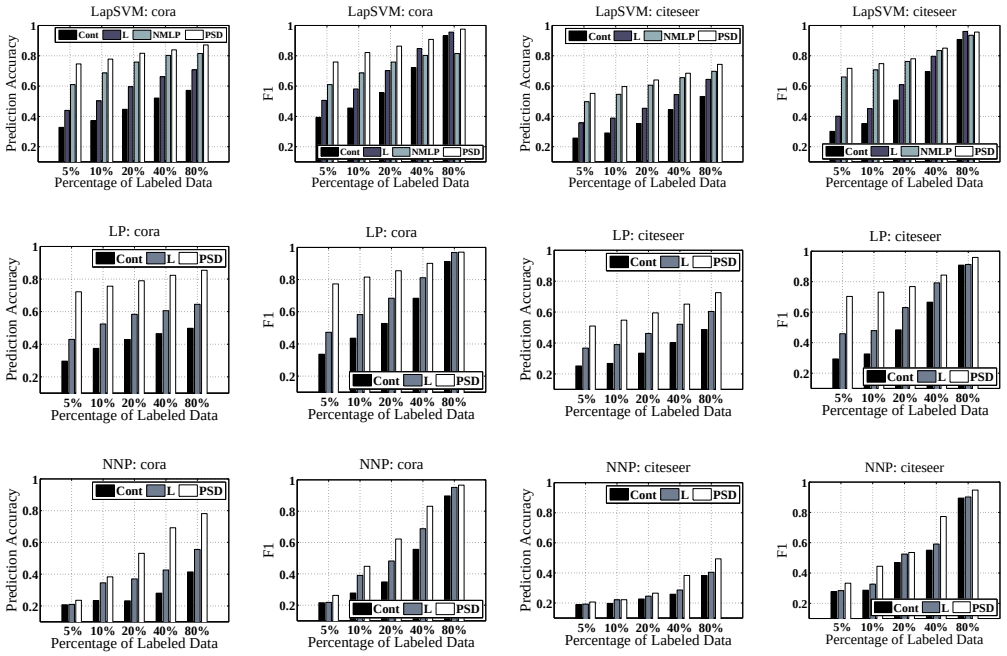


Fig. 3. Prediction accuracy and F1 measure comparison on *cora* and *citeseer* when $k = 3$ (each row for one GSSL approach; left two columns for *cora* and right two columns for *citeseer*).

figuring out those important features that are more reasonable for distance measure between different instance pairs.

4.1.3. Attack Classification. Each attack in this dataset is classified into one of six possible classes, i.e., arson, bombing, kidnapping, NBCR, weapon attack, and other attack. These attacks are linked either because they occurred in the same location or they were organized by the same terrorist organization. Two different subsets of this data set are derived due to different linkage type as aforementioned.

GSSL approaches are applied over different graphs induced from Cont, $\mathcal{L}$, CC, CC', and PSD. Similarly, Table VI shows the average prediction accuracy when the label ratio is fixed to 20% and k varies from $\{1, 3, 5, 7\}$, and Fig. 4 depicts the performance comparisons when there are different numbers of labeled data. It should be noted that Fig. 4 rules the performance of $\mathcal{L}$, CC, CC' and NMLP out, due to their relatively worse accuracy in Table VI. The advantage of PSD can be observed from both Table VI and Fig. 4, no matter what kind of GSSL approach is applied, how many nearest neighbors are considered or how much the labeled data are used.

In addition, it can be easily observed from the performance of NMLP and those GSSL approaches applied directly over $\mathcal{L}$, physical linkages for these two datasets are noisy for classification. In fact, different kinds of terrorist attacks may occur in the same location and one terrorist organization may organize different kinds of attacks. However, by introducing different distance metrics for different instance pairs, PSD can extract useful information from these noisy physical linkages to facilitate classification.

Furthermore, from Table VI, we can find that CC can provide better prediction performance than $\mathcal{L}$. However, it is not as good as Cont which is only based on the data attributes. One possible reason is that the total number of labeled data is too small, since the dataset size is small and only 20% data are labeled. The learning of CC is

Table VI. Accuracy comparison on *ta-loc* and *ta-loc-org* with fixed label ratio 20%. The best performance and its comparable performances are bolded (statistical significance examined via pairwise t-tests at 95% confidence level).

Data	Method	Metric	$k=1$	$k=3$	$k=5$	$k=7$
ta-loc	LapSVM	Cont	.767±.016	.793±.015	.790±.015	.795±.015
		$\mathcal{L}$	.561±.033	.571±.034	.583±.035	.587±.034
		$\mathcal{CC}$	.682±.101	.678±.102	.661±.106	.653±.105
		$\mathcal{CC}'$	.592±.043	.591±.047	.580±.042	.572±.031
		PSD	**.777±.015**	**.805±.016**	**.801±.015**	**.805±.014**
	LP	Cont	.759±.018	.780±.017	.791±.013	.800±.016
		$\mathcal{L}$	.385±.022	.392±.023	.401±.024	.403±.026
		$\mathcal{CC}$	.684±.101	.684±.103	.673±.110	.667±.107
		$\mathcal{CC}'$	.583±.036	.570±.024	.567±.022	.562±.018
		PSD	**.768±.017**	**.791±.015**	**.803±.014**	**.811±.015**
	NNP	Cont	.482±.082	.695±.049	.719±.049	.739±.039
		$\mathcal{L}$	.242±.022	.245±.029	.166±.024	.120±.027
		$\mathcal{CC}$	.295±.127	.581±.095	.548±.099	.531±.098
		$\mathcal{CC}'$	**.544±.023**	.579±.052	.560±.015	.558±.014
		PSD	.516±.068	**.709±.035**	**.735±.041**	**.756±.030**
	NMLP	-	.337±.049	.321±.047	.321±.045	.301±.061
	ISD	-	.711±.026	.757±.022	.780±.023	.795±.019
	LapSVM	PSD	**.777±.015**	**.805±.016**	**.801±.015**	**.805±.014**
ta-loc-org	LapSVM	Cont	.722±.045	.805±.022	.814±.024	.811±.026
		$\mathcal{L}$	.649±.035	.663±.029	.666±.029	.664±.028
		$\mathcal{CC}$	.547±.090	.614±.110	.611±.106	.612±.109
		$\mathcal{CC}'$	.541±.105	.610±.074	.612±.084	.614±.085
		PSD	**.743±.039**	**.817±.021**	**.826±.023**	**.821±.024**
	LP	Cont	.554±.036	.776±.032	.790±.024	.808±.021
		$\mathcal{L}$	.353±.042	.366±.042	.366±.042	.366±.042
		$\mathcal{CC}$	.353±.079	.618±.086	.628±.100	.622±.115
		$\mathcal{CC}'$	.271±.103	.581±.064	.597±.069	.602±.079
		PSD	**.572±.037**	**.788±.030**	**.800±.023**	**.819±.021**
	NNP	Cont	.731±.048	.770±.039	.787±.041	.763±.049
		$\mathcal{L}$	.357±.044	.480±.022	.515±.002	.515±.000
		$\mathcal{CC}$	.596±.067	.568±.069	.546±.084	.534±.068
		$\mathcal{CC}'$	.620±.071	.591±.060	.600±.101	.576±.113
		PSD	**.751±.045**	**.781±.041**	**.798±.038**	**.774±.048**
	NMLP	-	.560±.050	.495±.064	.430±.073	.379±.083
	ISD	-	**.764±.032**	.793±.032	.805±.025	.807±.028
	LapSVM	PSD	.743±.039	**.817±.021**	**.826±.023**	**.821±.024**

probably unreliable due to limited training instances and class-imbalance problem between $y'=1$ (with the same class) and $y'=0$ (from different classes). Moreover, we expect that by treating linked and unlinked instances separately, CC' could be more appropriate. However, it should be noticed that if we further separate labeled data, the learning could be more unreliable. Therefore, as shown in Table VI, CC' is not helpful in most cases compared to CC.

4.1.4. Webpage Categorization. The WebKB dataset contains webpages from four isolated computer science departments, categorized into topics as course, faculty, student, project and staff. Similarly, Table VII shows the average prediction accuracy when the label ratio is fixed to 20% while k changes, and Fig. 5 depicts the performance comparisons when there are different numbers of labeled data. Note that for report convenience, we rule out the results with $k=1$. Although Fig. 5 shows that PSD can improve the prediction performance when $k=3$, no matter how much the number

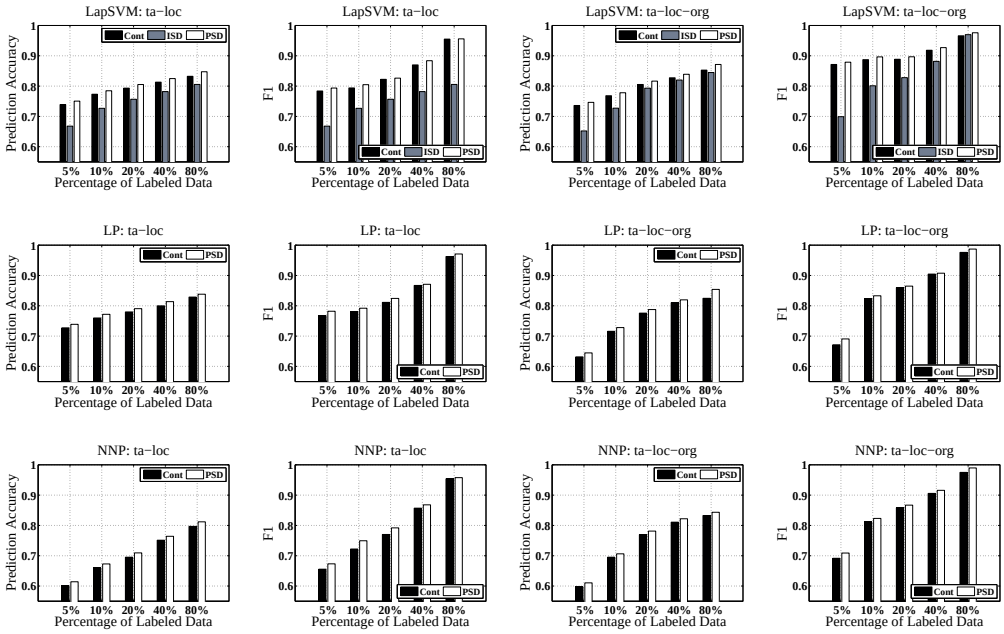


Fig. 4. Prediction accuracy and F1 measure comparison on *ta-loc* and *ta-loc-org* when $k = 3$ (each row for one GSSL approach; left two columns for *ta-loc* and right two columns for *ta-loc-org*).

of labeled data is, results in Table VII with different k's show that ISD and GSSL approaches over CC or CC' sometimes can provide better learning performance.

Compared to the performance on *Terrorist Attack*, CC and CC' are sometimes helpful. The possible reason is that the comparatively higher feature dimension as indicated in Table II makes the learning of CC and CC' reliable. However, contradict results on different data sets indicate that CC and CC' are not reliable or practically useful, not to mention their high time cost as summarized in Table IV.

From the performance of NMLP and GSSL approaches over $\mathcal{L}$, we can easily find that physical linkages for these four datasets are very noisy, which is reasonable because webpages from totally different classes may be linked by the hyperlinks as mentioned previously. However, GSSL approaches over PSD can still provide better prediction performance than over Cont or $\mathcal{L}$, and other approaches NMLP and ISD in almost all cases.

Overall, empirical evaluations on all used datasets confirm not only the usefulness of hidden information within existing physical linkages, but also our PSD proposal. By distinguishing different semantic meanings passed through different physical linkages, a more appropriate distance metric can be obtained and further help improve the classification performance. Besides, different extent of advantages on different number of labeled data as in Fig. 3, confirm our concern that if labeled data themselves are sufficient enough for learning, considering additional information as physical linkages is not necessary, although it can be helpful. Moreover, significantly better prediction performance on *cora* and *citeseer* tells that the helpfulness of PSD is more significant when data attributes are insufficient to distinguish instances from different classes.

Table VII. Accuracy comparison on *WebKB* (*i.e.*, *cornell*, *texas*, *washington* and *wisconsin*) with fixed label ratio 20%. The best performance and its comparable performances are bolded (statistical significance examined via pairwise t-tests at 95% confidence level).

Method	Metric	$k=3$	$k=5$	$k=7$	$k=3$	$k=5$	$k=7$
		cornell			*texas*		
LapSVM	Cont	.536±.043	.498±.026	.473±.021	.607±.028	**.573±.009**	**.563±.003**
	$\mathcal{L}$	.348±.067	.349±.069	.349±.069	.456±.117	.459±.114	.459±.113
	$\mathcal{CC}$	.500±.055	**.534±.050**	**.536±.051**	.534±.030	.560±.003	562±.000
	$\mathcal{CC}'$	.429±.000	.428±.001	.429±.000	.562±.000	.562±.000	.562±.000
	PSD	**.546±.030**	.509±.021	.483±.017	**.615±.023**	**.573±.009**	**.563±.002**
LP	Cont	.557±.049	.523±.047	.499±.042	.601±.042	.594±.043	.565±.021
	$\mathcal{L}$	.320±.078	.322±.076	.322±.076	.401±.138	.398±.136	.397±.135
	$\mathcal{CC}$	.486±.053	.526±.053	**.531±.053**	.532±.030	.561±.003	.562±.000
	$\mathcal{CC}'$	.429±.000	.429±.000	.429±.000	.562±.000	.562±.000	.562±.000
	PSD	**.567±.049**	**.533±.041**	.509±.033	**.609±.042**	**.603±.044**	**.566±.021**
NNP	Cont	.383±.101	.400±.092	.423±.040	.423±.185	.451±.182	**.562±.000**
	$\mathcal{L}$	.213±.018	.213±.006	.213±.003	.252±.140	.424±.187	.549±.070
	$\mathcal{CC}$	**.420±.105**	**.472±.105**	**.517±.067**	.348±.138	.546±.072	**.562±.000**
	$\mathcal{CC}'$	.348±.100	.370±.090	.406±.061	**.562±.000**	**.562±.000**	**.562±.000**
	PSD	.403±.098	.414±.091	.434±.040	424±.175	.458±.176	**.562±.000**
NMLP	-	.371±.050	.369±.059	.360±.077	.512±.048	.473±.083	.351±.118
ISD	-	.513±.049	.505±.044	**.491±.046**	.547±.117	**.580±.081**	**.584±.031**
LapSVM	PSD	**.546±.030**	**.509±.021**	.483±.017	**.615±.023**	.573±.009	.563±.002
		washington			*wisconsin*		
LapSVM	Cont	.700±.033	.696±.035	.659±.045	.649±.024	.634±.016	.619±.019
	$\mathcal{L}$	.446±.082	.452±.080	.450±.083	.407±.058	.407±.055	.410±.053
	$\mathcal{CC}$	.627±.062	.700±.031	**.716±.024**	.391±.035	.422±.031	.440±.026
	$\mathcal{CC}'$	.488±.009	.488±.009	.488±.009	.469±.011	.467±.009	.467±.010
	PSD	**.710±.034**	**.707±.032**	671±.040	**.659±.025**	**.643±.017**	**.629±.020**
LP	Cont	.708±.037	.670±.053	.665±.068	.644±.031	.620±.027	.604±.033
	$\mathcal{L}$	.385±.107	.381±.105	.380±.105	.379±.071	.374±.070	.376±.071
	$\mathcal{CC}$	.603±.063	**.687±.027**	**.705±.021**	.388±.043	.418±.038	.437±.026
	$\mathcal{CC}'$	.477±.000	.477±.000	.477±.000	.461±.007	.458±.003	.457±.001
	PSD	**.717±.037**	.679±.053	.676±.067	**.653±.030**	**.629±.028**	**.613±.034**
NNP	Cont	.477±.167	.475±.137	.456±.087	**.562±.096**	.528±.031	.554±.036
	$\mathcal{L}$	.274±.015	.280±.003	.279±.000	.302±.059	.247±.106	.249±.118
	$\mathcal{CC}$	.434±.085	**.507±.127**	**.584±.119**	.313±.038	.415±.051	.365±.067
	$\mathcal{CC}'$	.478±.037	.470±.036	.477±.000	.402±.078	.450±.032	.446±.052
	PSD	**.490±.161**	.487±.126	.472±.084	**.562±.096**	**.538±.030**	**.563±.036**
NMLP	-	.455±.036	.455±.055	.381±.082	.447±.029	.453±.032	.418±.056
ISD	-	.554±.120	.580±.092	.585±.070	.609±.053	.576±.047	.580±.044
LapSVM	PSD	**.710±.034**	**.707±.032**	**.671±.040**	**.659±.025**	**.643±.017**	**.629±.020**

4.2. Extension to Multi-Label Scenario

In this section, we conduct an extension to the image annotation dataset of MSRA-MM [Li et al. 2009] database. MSRA-MM database is collected by Microsoft Research Asia (MSRA) using Microsoft Live Search and all the labels are annotated by humans. In this paper, we use the 1st version which is collected in March 2009. Images in this dataset are collected from very different sources, e.g., logos for web sites, cartoons, movies, real scene images, etc. In this paper, we select all real scene images, and an image annotation data set of size 1,605 is finally obtained with thirty-eight labels, i.e., *airplane, animal, baby, beach, bike, bird, boat, building, bus, candle, car, cat, cattle, cloud, desert, dog, dolphin, elephant, fire, fireworks, horse, ice, jungle, landscape, leaf, lightning, mountains, penguin, people, rock, sea, ship, sky, sun, swimming, water, wa-*

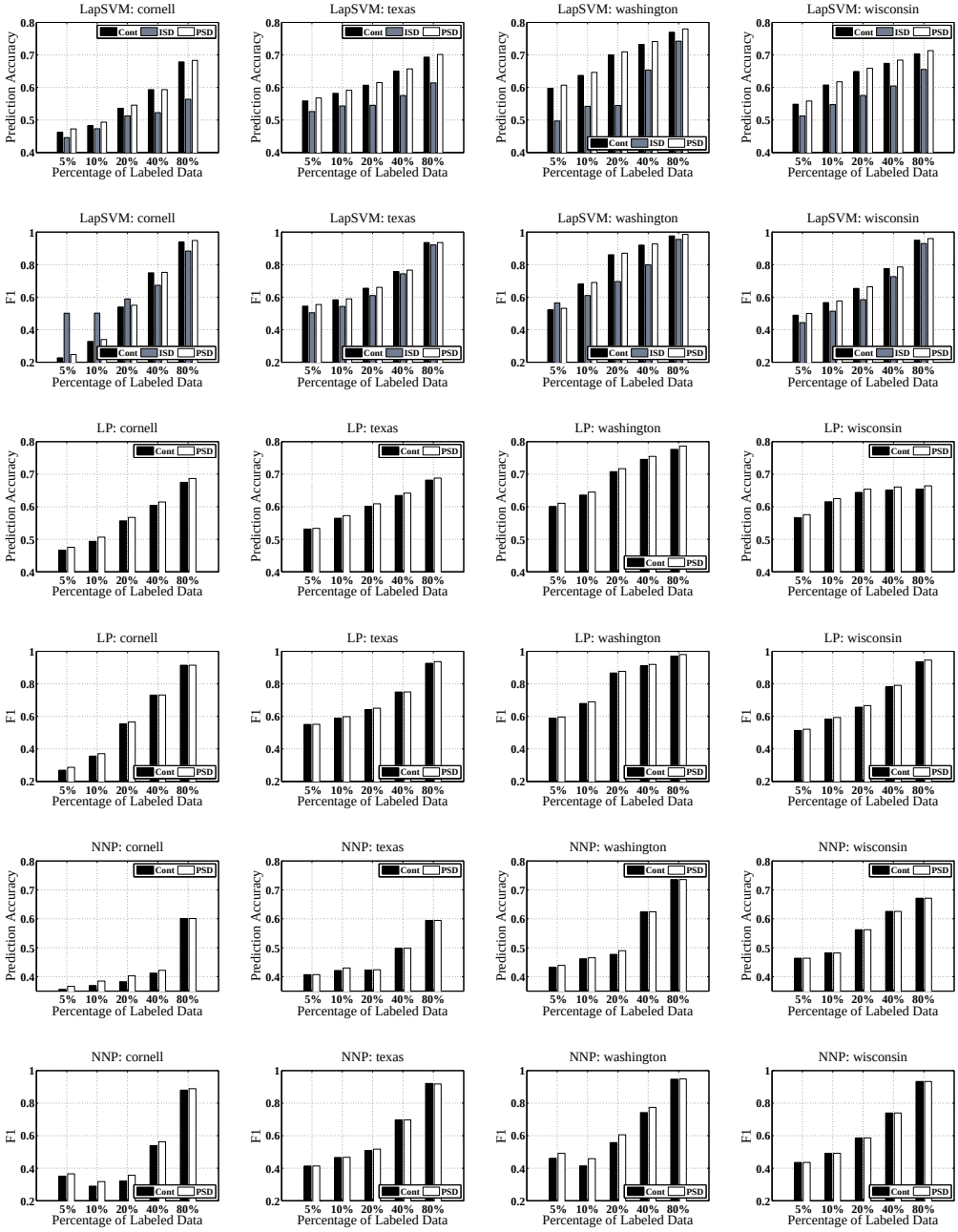


Fig. 5. Prediction accuracy and F1 measure comparison on *WebKB* when $k = 3$ (each column for one sub dataset from left to right: *cornell*, *texas*, *washington*, *wisconsin*).

terfall, woman. Around 92% of these images are with more than one label and there are at most 11 labels annotated to one image. The average number of labels for each image is 3.85 ± 1.75. The physical linkages are manually added if two images have the same

label. In this case, 916,364 linkages are gathered. It is common in social networks that two persons knowing each other may have not added each other as friends online. As a simulation, we conduct random sampling to select 10,000 linkages from them, and each node is constrained to be within at least one linkage for empirical study. As a result, the average number of linkages for each node becomes 12.46 ± 4.87.

To visualize the distinguished semantic meanings passed through different linkages, we conduct image segmentation following the techniques used in the study [Wang et al. 2001]. Then, each image is represented by a bag of at most sixteen 6-dimensional instances, denoted as $B_i = \{\mathbf{x}_{ij} | j = 1, \cdots, N_i\}$ where N_i is the total number of segmentations of this image. After clustering on all images' instances by k-means clustering with k set to 2,000, we get 1,897 instance prototypes for the whole data set as $\{\mathbf{p}_1, \cdots, \mathbf{p}_{1897}\}$. Finally, we map each image bag onto these prototypes by finding the minimum Euclidean distance from all instances of this bag to each prototype as $\phi_q(B_i) = \min_{j=1,\cdots,N_i}((\mathbf{x}_{ij} - \mathbf{p}_q)^\top (\mathbf{x}_{ij} - \mathbf{p}_q))^{\frac{1}{2}}$, respectively. A similar technique has been used in studies [Zhou and Zhang 2007; Zhou et al. 2012; Li et al. 2012]. Now, each image is represented by the vector as $[\psi_1, \cdots, \psi_{1897}]$, where $\psi_q = \exp(-\phi_q(B_i)^2/\sigma^2)$ and σ is the mean distance. As a result, a multi-label dataset with 1,605 images, each of which has 1,897 features, is obtained. More importantly, we can track the semantic meaning for each feature according to its corresponding prototype.

First, we randomly conduct 10 trials with the widely used multi-label approach ML-kNN [Zhang and Zhou 2007] applied over graphs induced by Cont, $\mathcal{L}$, and PSD. The parameter k is set from $\{1, 3, 5\}$, while other parameters are set as default. With the label ratio changing from the set $\{5\%, 10\%, 20\%, 40\%, 80\%\}$, the performance is evaluated with five popular multi-label measurements, i.e., average precision (a.p.), coverage (co.), hamming loss (h.l.), one error (o.e.), and ranking loss (r.l.). For detailed descriptions about these measurements, refer to the study [Schapire and Singer 2000]. As shown in Table VIII, the prediction performance with our PSD is significantly better than others when there are fewer labeled data. It is reasonable that when there are 80% labeled data, the performance with PSD and Cont are comparable, since training data themselves are sufficient to learn a good classifier. Besides, it can be easily seen that directly using linkages $\mathcal{L}$ may lead to inappropriate conclusions about the data.

We also compare the prediction performance of ML-kNN with our PSD to W_{ELL} approach, which was proposed in study [Sun et al. 2010] to handle the weak-label problem. We find that W_{ELL} could not handle this multi-label problem when there are only 5%, 10% and 20% labeled data due to some optimization errors. Therefore, as summarized in Table IX, the prediction performance with 40% and 80% labeled data is compared. In addition to those five popular multi-label measurements, we employ the measurements Macro-F1 (ma.) and Micro-F1 (mi.) as in the work [Sun et al. 2010]. Results show that ML-kNN with PSD is a little bit higher on *hamming loss* and *one error*, whereas it is significantly better than W_{ELL} on all other measurements.

It is noteworthy that the time cost of learning PSD for the multi-label scenario may be much higher than that of the multi-class scenario due to the propagation over the whole graph in the second stage. In fact, we find that the average time for learning PSD per trial is 151.13 minutes, which is still acceptable.

Then we randomly choose five images as shown in Fig. 6(a) to visualize the semantic meanings distinguished through different linkages during the PSD$_{ml}$ learning, where images (1), (2) and (3) form a clique and images (3), (4) and (5) compose a non-metric linkage, which can also be observed by their true labels.

First, for the *Clique(123)*, intuitively those linkages $(1)(2)$, $(2)(3)$ and $(1)(3)$ should share the semantic meanings: cloud, sky, and leaf. From the learned pairwised distance metrics, i.e., $\mathbf{w}_{12}$, $\mathbf{w}_{23}$ and $\mathbf{w}_{13}$, we match the two largest weighted prototypes for each

Table VIII. ML-kNN performance comparison with different metrics. ↑ indicates "the larger, the better"; ↓ indicates "the smaller, the better". The best performance and its comparable performances are bolded (statistical significance examined via pairwise t-tests at 95% confidence level).

k	Ratio	Metric	*a.p.*↑	*co.*↓	*h.l.*↓	*o.e.*↓	*r.l.*↓
$k=1$	5%	$Cont$	.605±.005	10.538±.057	**.100±.000**	.457±.011	.113±.001
		$\mathcal{L}$	.576±.009	11.189±.217	.102±.002	.539±.030	.113±.003
		PSD	**.614±.006**	**10.527±.058**	**.100±.000**	**.447±.012**	**.104±.001**
	10%	$Cont$	.635±.004	10.135±.048	**.095±.000**	.374±.006	**.101±.001**
		$\mathcal{L}$	.594±.005	10.677±.087	.101±.000	.480±.020	.105±.001
		PSD	**.644±.004**	**10.126±.051**	**.095±.000**	**.365±.005**	**.101±.001**
	20%	$Cont$	.690±.004	9.513±.036	.084±.000	.248±.013	**.086±.000**
		$\mathcal{L}$	.623±.006	10.456±.095	.101±.000	.372±.019	.099±.001
		PSD	**.699±.005**	**9.504±.035**	**.083±.000**	**.239±.013**	**.086±.000**
	40%	$Cont$	.783±.003	8.186±.051	.063±.000	.111±.006	**.061±.000**
		$\mathcal{L}$	.646±.005	10.241±.079	.101±.000	.323±.010	.093±.001
		PSD	**.793±.004**	**8.176±.051**	**.062±.000**	**.102±.006**	**.061±.000**
	80%	$Cont$	**.934±.002**	**4.864±.028**	**.021±.000**	**.017±.003**	**.019±.000**
		$\mathcal{L}$	.662±.004	9.843±.038	.084±.000	.286±.010	.085±.000
		PSD	**.934±.002**	**4.864±.028**	**.021±.000**	**.017±.003**	**.019±.000**
$k=3$	5%	$Cont$	.567±.006	10.624±.123	.099±.000	.461±.026	.110±.002
		$\mathcal{L}$	.536±.016	11.808±.489	.100±.000	.479±.025	.129±.010
		PSD	**.576±.006**	**10.622±.124**	**.098±.000**	**.452±.029**	**.100±.001**
	10%	$Cont$	.588±.007	10.194±.076	.097±.000	.429±.020	.101±.002
		$\mathcal{L}$	.581±.006	11.038±.094	.101±.000	**.401±.017**	.112±.002
		PSD	**.600±.005**	**10.185±.078**	**.096±.000**	.413±.016	**.091±.001**
	20%	$Cont$	.632±.007	9.588±.053	.094±.000	.355±.021	.090±.001
		$\mathcal{L}$	.620±.004	10.359±.079	.100±.000	**.334±.016**	.099±.000
		PSD	**.643±.007**	**9.573±.057**	**.093±.000**	.343±.019	**.089±.001**
	40%	$Cont$	.695±.007	8.396±.068	.085±.000	.272±.012	.071±.001
		$\mathcal{L}$	.669±.003	9.805±.064	.099±.000	.277±.008	.086±.001
		PSD	**.705±.006**	**8.385±.068**	**.084±.000**	**.261±.011**	**.070±.001**
	80%	$Cont$	.808±.003	**5.704±.045**	**.060±.000**	.150±.006	.036±.000
		$\mathcal{L}$	.717±.003	9.043±.050	.078±.000	.193±.010	.072±.000
		PSD	**.817±.004**	5.707±.059	.061±.000	**.138±.005**	**.035±.000**
$k=5$	5%	$Cont$	.569±.010	10.745±.111	.100±.000	.412±.029	.112±.002
		$\mathcal{L}$	.530±.019	11.522±.478	.100±.001	.446±.024	.129±.009
		PSD	**.579±.011**	**10.736±.123**	**.099±.000**	**.402±.026**	**.111±.002**
	10%	$Cont$	.588±.009	10.265±.077	.099±.001	.391±.014	.103±.002
		$\mathcal{L}$	.572±.008	11.012±.229	.100±.000	.399±.023	.115±.004
		PSD	**.600±.007**	**10.256±.077**	**.098±.001**	**.386±.015**	**.102±.002**
	20%	$Cont$	.627±.005	9.629±.087	.097±.000	**.339±.009**	.092±.002
		$\mathcal{L}$	.621±.003	10.357±.110	.099±.000	**.339±.010**	.100±.001
		PSD	**.638±.007**	**9.612±.077**	**.096±.000**	.340±.012	**.091±.002**
	40%	$Cont$	.679±.004	8.529±.066	.089±.000	.276±.005	.075±.001
		$\mathcal{L}$	.680±.005	9.538±.060	.097±.001	**.254±.009**	.082±.001
		PSD	**.691±.003**	**8.510±.078**	**.088±.000**	.274±.009	**.074±.001**
	80%	$Cont$	.778±.002	6.208±.044	.066±.000	.172±.006	.043±.000
		$\mathcal{L}$	.745±.004	8.589±.048	.074±.000	**.146±.008**	.066±.000
		PSD	**.789±.002**	**6.201±.055**	**.065±.000**	.170±.004	**.042±.000**

distance metric to these three images, respectively. As shown in Fig. 6(b), we find that these three linkages share the semantic meanings *cloud, sky* completely with each other. Although the semantic meaning *leaf* should be shared by these three linkages,

Table IX. Performance comparison with W_{ELL}, $\uparrow$ indicates "the larger, the better"; $\downarrow$ indicates "the smaller, the better". The best performance and its comparable performances are bolded (statistical significance examined via pairwise t-tests at 95% confidence level).

Ratio	40%		80%	
Method	W_{ELL}	$ML\text{-}kNN_{PSD}$	W_{ELL}	$ML\text{-}kNN_{PSD}$
a.p.$\uparrow$	.542±.004	**.793±.004**	.854±.002	**.934±.002**
co.$\downarrow$	21.983±.137	**8.176±.051**	11.580±.080	**4.864±.028**
h.l.$\downarrow$	**.061±.000**	.062±.000	**.020±.000**	.021±.000
o.e.$\downarrow$	**.000±.000**	.101±.006	**.000±.000**	.017±.003
r.l.$\downarrow$	.597±.004	**.061±.000**	.198±.002	**.019±.000**
ma.$\uparrow$	.514±.004	**.574±.004**	.866±.002	**.883±.002**
mi.$\uparrow$	.574±.000	**.592±.002**	.891±.000	**.896±.000**

PSD_{ml} empirically finds different regions shared between different pairs. Actually, this is the purpose of PSD that distinguishes different linking reasons for different instance pairs.

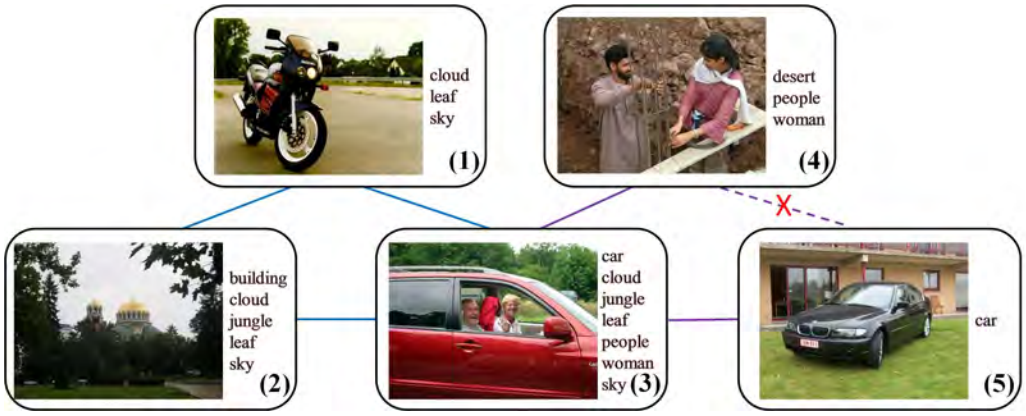
As depicted in Fig. 6(c) for *NonM(345)*, we also match the two largest weighted prototypes for each linkage to these images according to the pairwised distance metrics $\mathbf{w}_{34}$ and $\mathbf{w}_{35}$, respectively. It can be easily observed that linkages $(3)(4)$ and $(3)(5)$ pass totally different semantic meanings as we expected. It should be noted that since image (3) and image (5) only share the content *car*, it is reasonable that the two largest weighted prototypes for linkage $(3)(5)$ are different parts of the *car*.

5. CONCLUSIONS

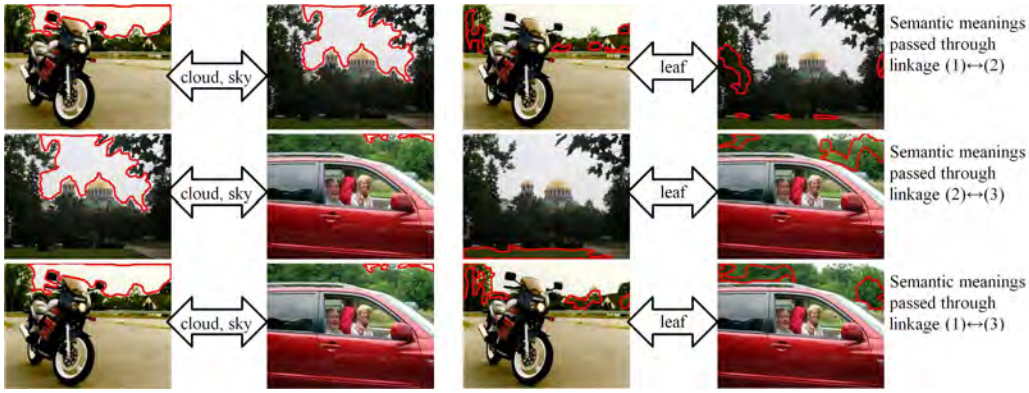
Distance metric learning has received much attention during the past decade. Although many effective algorithms have been developed, few tried to exploit physical link information naturally existing in many real tasks. Moreover, when there are only limited labeled data or no additional constraints are available, current distance metric learning techniques can hardly lead to good performance.

By noting that data points in real tasks are often connected by physical links, in this paper we propose the PSD approach which tries to distinguish different semantic meanings passed through different linkages. PSD is able to work even when there are very limited labeled training data points and no explicit constraints given; this owes to its exploitation of the structure of physical linkages, particularly the key observations that non-metric and clique linkages imply the appearance of different or unique semantic meanings, respectively. We formulate the PSD learning process into an optimization problem that can be solved effectively. The usefulness of PSD is empirically verified in both multi-class learning and multi-label learning on a broad range of datasets.

One interesting future work is to improve our PSD approach for even larger real world applications, particularly in multi-label scenario. In our current experimental setting, the physical linkages are simulated by considering the shared labels between different multi-label instances. In fact, linkage information can also be obtained by considering information such as images taken by the same person, at the same date or at the same place; such considerations lead to more experimental data sets for future studies. Another insight direction is to categorize the relationships between nodes by exploiting the edge label information.



(a) Images with ground-truth labels and physical linkages



(b) Semantic patterns shared by three linkages within $Clique(123)$

(c) Different semantic patterns passed through different linkages within $NonM(345)$

Fig. 6. Examples of semantic patterns possessed by different linkages (Images are from *MSRA-MM* data set; Patterns are circled by red line).

ACKNOWLEDGMENTS

The authors would like to thank Microsoft Research Asia for providing the image dataset and wish to thank the editor and anonymous reviewers for their helpful comments and suggestions. The authors would also like to thank Nan Li, and Shao-Yuan Li for proof reading the article.

REFERENCES

BELKIN, M., NIYOGI, P., AND SINDHWANI, V. 2006. A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research 7*, 2399–2434.

BEZDEK, J. C. AND HATHAWAY, R. J. 2003. Convergence of alternating optimization. *Neural, Parallel and Scientific Computations 11*, 351–368.

BLUM, A. AND CHAWLA, S. 2001. Learning from labeled and unlabeled data using graph mincuts. In *Proceedings of the 18th International Conference on Machine Learning*. San Francisco, CA, 19–26.

FROME, A., SINGER, Y., AND MALIK, J. 2007a. Image retrieval and classification using local distance functions. In *Advances in Neural Information Processing Systems 19*. 417–424.

FROME, A., SINGER, Y., SHA, F., AND MALIK, J. 2007b. Learning globally-consistent local distance functions for shape-based image retrieval and classification. In *Proceedings of the 11th International Conference on Computer Vision*. Rio de Janeiro, Brazil, 1–8.

GENG, X., ZHAN, D.-C., AND ZHOU, Z.-H. 2005. Supervised nonlinear dimensionality reduction for visualization and classification. *IEEE Transactions on Systems, Man, and Cybernetics - Part B 35*, 1098–1107.

GOLDBERGER, J., ROWEIS, S., HINTON, G., AND SALAKHUTDINOV, R. 2005. Neighbourhood components analysis. In *Advances in Neural Information Processing Systems 17*. 513–520.

HUA, X.-S. AND QI, G.-J. 2008. Online multi-label active annotation: Towards large-scale content-based video search. In *Proceeding of the 16th ACM International Conference on Multimedia*. Vancouver, Canada, 141–150.

JIN, R., WANG, S., AND ZHOU, Z.-H. 2009. Learning a distance metric from multi-instance multi-label data. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Miami, FL, 896–902.

KANG, F., JIN, R., AND SUKTHANKAR, R. 2006. Correlated label propagation with application to multi-label learning. In *Proceedings of the Computer Society Conference on Computer Vision and Pattern Recognition*. New York, NY, 1719–1726.

KONG, X., NG, M. K., AND ZHOU, Z.-H. 2013. Transductive multilabel learning via label set propagation. *IEEE Transactions on Knowledge and Data Engineering 25*, 704–719.

KUMAR, N. AND KUMMAMURU, K. 2007. Semi-supervised clustering with metric learning using relative comparisons. *IEEE Transactions on Knowledge and Data Engineering 20*, 496–503.

KWOK, J. AND TSANG, I. 2003. Learning with idealized kernels. In *Proceedings of the 20th International Conference on Machine Learning*. Washington, DC, 400–407.

LI, H., WANG, M., AND HUA, X.-S. 2009. MSRA-MM 2.0: A large-scale web multimedia dataset. In *Proceedings of the International Conference on Data Mining Workshops*. Miami, FL, 164–169.

LI, Y.-F., HU, J., JIANG, Y., AND ZHOU, Z.-H. 2012. Towards discovering what patterns trigger what labels. In *Proceedings of the 26th AAAI Conference on Artifical Intelligence*. Toronro, Canada, 1012–1018.

LI, Z., LIU, J., AND TANG, X. 2008. Pairwise constraint propagation by semidefinite programming for semi-supervised classification. In *Proceedings of the 25th International Conference on Machine Learning*. Helsinki, Finland, 576–583.

MCELIECE, L. K., GUPTA, K. M., AND AHA, D. W. 2007. Cautious inference in collective classification. In *Proceedings of the 22nd AAAI Conference on Artficial Intelligence*. Vancouver, Canada, 596–601.

MELACCI, S. AND BELKIN, M. 2011. Laplacian support vector machines trained in the primal. *Journal of Machine Learning Research 12*, 1149–1184.

MENG, W., YANG, L., AND HUA, X.-S. 2009. Msra-mm: Bridging research and industrial societies for multimedia information retrieval. Tech. Rep. MSR-TR-2009-30, Microsoft.

NAMATA, G. M., SEN, P., BILGIC, M., AMD B. GALLAGHER, L. G., AND ELIASSI-RAD, T. 2009. Collective classification for text classification. In *Text Mining: Classification, Clustering, and Applications*, M. Sahami and A. Srivastava, Eds. Taylor and Francis, Chapter 3, 51–69.

NEVILLE, J. AND JENSEN, D. 2000. Interative classification in relational data. In *Proceedings of AAAI Workshop on Statistical Relational Learning*. Austin, TX, 13–20.

SCHAPIRE, R. E. AND SINGER, Y. 2000. BoosTexter: A boosting-based system for text categorization. *Machine Learning 39,* 2-3, 135–168.

SEN, P. AND GETOOR, L. 2007. Link-based classification. Tech. Rep. CS-TR-4858, University of Maryland.

SEN, P., NAMATA, G. M., BILGIC, M., GETOOR, L., GALLAGHER, B., AND ELIASSI-RAD, T. 2008. Collective classification in network data. *AI Magazine 29*, 93–106.

SUN, Y.-Y., ZHANG, Y., AND ZHOU, Z.-H. 2010. Multi-label learning with weak label. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*. Atlanta, GA, 593–598.

TAN, X., CHEN, S., ZHOU, Z.-H., AND LIU, J. 2006. Learning non-metric partial similarity based on maximal margin criterion. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. New York, NY, 138–145.

TAN, X., CHEN, S., ZHOU, Z.-H., AND LIU, J. 2009. Face recognition under occlusions and variant expressions with partial similarity. *IEEE Transactions on Information Forensics and Security 4*, 217–230.

WANG, J., LI, J., AND WIEDERHOLD, G. 2001. Simplicity: Semantics sensitive integrated matching for picture libraries. *IEEE Transactions on Pattern Analysis and Machine Intelligence 23*, 947–963.

WANG, J., POHLMEYER, E., HANNA, B., JIANG, Y.-G., SAJDA, P., AND CHANG, S.-F. 2009. Brain state decoding for rapid image retrieval. In *Proceeding of the 17th ACM International Conference on Multimedia*. Beijing, China, 945–954.

WANG, J., ZHAO, Y., WU, X., AND HUA, X.-S. 2011. A transductive multi-label learning approach for video concept detection. *Pattern Recognition 44,* 10-11, 2274–2286.

WEINBERGER, K. Q., BLITZER, J., AND SAUL, L. K. 2005. Distance metric learning for large margin nearest neighbor classification. In *Advances in Neural Information Processing Systems 17*. 1473–1480.

WU, G., CHANG, E. Y., AND ZHANG, Z. 2005. Learning with non-metric proximity matrices. In *Proceedings of the 13th Annual ACM International Conference on Multimedia*. Singapore, 411–414.

WU, L., YING, X., WU, X., AND ZHOU, Z.-H. 2011. Line orthogonality in adjacency eigenspace with application to community partition. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*. Barcelona, Spain, 2349–2354.

XIANG, S., NIE, F., AND ZHANG, C. 2008. Learning a mahalanobis distance metric for data clustering and classification. *Pattern Recognition 41*, 3600–3612.

XING, E., NG, A., JORDAN, M., AND RUSSELL, S. 2003. Distance metric learning with application to clustering with side-information. In *Advances in Neural Information Processing Systems 15*. 505–512.

YANG, L. AND JIN, R. 2006. Distance metric learning: A comprehensive survey. *Michigan State University 2*.

YANG, S.-J., JIANG, Y., AND ZHOU, Z.-H. 2013. Multi-instance multi-label learning with weak label. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*. Beijing, China.

YEUNG, D. AND CHANG, H. 2007. A kernel approach for semi-supervised metric learning. *IEEE Transactions on Neural Networks 18*, 141–149.

ZHAN, D.-C., LI, M., LI, Y.-F., AND ZHOU, Z.-H. 2009. Learning instance specific distances using metric propagation. In *Proceedings of the 26th International Conference on Machine Learning*. Montreal, Canada, 1225–1232.

ZHANG, M.-L. AND ZHOU, Z.-H. 2007. ML-kNN: A lazy learning approach to multi-label learning. *Pattern Recognition 40*, 2038–2048.

ZHANG, W., LU, Y., XUE, X., AND FAN, J. 2011. Automatic image annotation with weakly labeled dataset. In *Proceeding of the 19th ACM International Conference on Multimedia*. Scottsdale, AZ, 1185–1188.

ZHANG, Y. AND ZHOU, Z.-H. 2009. Non-metric label propagation. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*. Pasadena, CA, 1357–1362.

ZHOU, D., BOUSQUET, O., LAL, T. N., WESTON, J., AND SCHÖLKOPF, B. 2004. Learning with local and global consistency. In *Advances in Neural Information Processing Systems 16*. 321–328.

ZHOU, Z.-H. AND ZHANG, M.-L. 2007. Multi-instance multi-label learning with application to scene classification. In *Advances in Neural Information Processing Systems 19*. 1609–1616.

ZHOU, Z.-H., ZHANG, M.-L., HUANG, S.-J., AND LI, Y.-F. 2012. Multi-instance multi-label learning. *Artificial Intelligence 176*, 2291–2320.

ZHU, X. AND GHAHRAMANI, Z. 2002. Learning from labeled and unlabeled data with label propagation. Tech. rep., Carnegie Mellon University.

ZHU, X., GHAHRAMANI, Z., AND LAFFERTY, J. 2003. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the 20th International Conference on Machine Learning*. Washington, DC, 912–919.