Pairwise Constraint Propagation by Semidefinite Programming for Semi-Supervised Classification

Zhenguo Li ZGLI5@IE.CUHK.EDU.HK
Jianzhuang Liu JZLIU@IE.CUHK.EDU.HK
Xiaoou Tang XTANG@IE.CUHK.EDU.HK
Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong

Abstract

We consider the general problem of learning from both pairwise constraints and unlabeled data. The pairwise constraints specify whether two objects belong to the same class or not, known as the must-link constraints and the cannot-link constraints. We propose to learn a mapping that is smooth over the data graph and maps the data onto a unit hypersphere, where two must-link objects are mapped to the same point while two cannot-link objects are mapped to be orthogonal. We show that such a mapping can be achieved by formulating a semidefinite programming problem, which is convex and can be solved globally. Our approach can effectively propagate pairwise constraints to the whole data set. It can be directly applied to multi-class classification and can handle data labels, pairwise constraints, or a mixture of them in a unified framework. Promising experimental results are presented for classification tasks on a variety of synthetic and real data sets.

1. Introduction

Learning from both labeled and unlabeled data, known as semi-supervised learning, has attracted considerable interest in recent years (Chapelle et al., 2006), (Zhu, 2005). The key to the success of semi-supervised learning is the cluster assumption (Zhou et al., 2004), stating that nearby objects and objects on the same manifold structure are likely to be in the same class. Different algorithms actually implement the cluster assumption from different viewpoints (Zhu et al.,), (Zhou et al., 2004), (Belkin et al., 2006), (Chapelle & Zien, 2005), (Zhang & Ando, 2006), (Szummer et al., 2002). When the cluster assumption is appropriate, we can properly classify the whole data set with only one labeled object for each class.

However, the distributions of real-world data are often more complex than expected, where there are circumstances that a class may consist of multiple separate groups or manifolds, and different classes may be close to or even overlap with each other. For example, a common experience is that face images of the same person under different poses and illuminations can be drastically different, while those with similar appearances may originate from two different persons. To handle the classification problems of such practical data, additional assumptions should be made and more supervisory information should be exploited when available.

Class labels of data are the most widely used supervisory information. In addition, pairwise constraints are also often seen, which specify whether two objects belong to the same class or not, known as the must-link constraints and the cannot-link constraints. Such pairwise constraints may arise from domain knowledge automatically or with a little human effort (Wagstaff & Cardie, 2000), (Klein et al., 2002), (Kulis et al., 2005), (Chapelle et al., 2006). They can also be obtained from data labels where objects with the same label are must-link while objects with different labels are cannot-link. Generally, we cannot infer data labels from only pairwise constraints, especially for multi-class data. In this sense, pairwise constraints are inherently weaker and thus more general than labels of data. Pairwise constrains have been widely used in the contexts of clustering with side information (Wagstaff et al., 2001), (Klein et al., 2002), (Xing et al., 2003), (Kulis et al., 2005), (Kamvar et al., 2003), (Globerson & Roweis, 2006), (Basu et al., 2004), (Bilenko

Appearing in *Proceedings of the 25th International Conference on Machine Learning*, Helsinki, Finland, 2008.

et al., 2004), (Bar-Hillel et al., 2003), (Hoi et al., 2007), where it has been shown that the presence of appropriate pairwise constraints can often improve the clustering performance.

In this paper, we consider a more general problem of semi-supervised classification from pairwise constraints and unlabeled data, which includes the traditional semi-supervised classification as a subproblem that considers labeled and unlabeled data. Note that the label propagation techniques, which are often used in the traditional semi-supervised classification (Zhou et al., 2004), (Zhu et al.,), (Belkin et al., 2006), cannot be readily generalized to propagate pairwise constraints. Recently, two methods (Goldberg et al., 2007), (Tong & Jin, 2007) are proposed to incorporate dissimilarity information in semi-supervised classification, which is similar to the cannot-link constraints. It is important to notice that the similarities between objects are not identical to the must-link constraints imposed on them. The former reflects their distances in the input space while the latter is often obtained using domain knowledge or specified by the user.

We propose an approach, called *pairwise constraint propagation* (PCP), that can effectively propagate pairwise constraints to the whole data set. PCP intends to learn a mapping that is smooth over the data graph and maps the data onto a unit hypersphere, where two must-link objects are mapped to the same point while two cannot-link objects are mapped to be orthogonal. Such a mapping can be implicitly achieved using the kernel trick via semidefinite programming, which is convex and can be solved globally. Our approach can be directly applied to multi-class classification and can handle data labels, pairwise constraints, or a mixture of them in a unified framework.

2. Motivation

Given a data set of n objects $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$ and two sets of pairwise must-link and cannot-link constraints, denoted respectively by $\mathcal{M} = \{(\mathbf{x}_i, \mathbf{x}_j)\}$ where $\mathbf{x}_i$ and $\mathbf{x}_j$ should be in the same class and $\mathcal{C} = \{(\mathbf{x}_i, \mathbf{x}_j)\}$ where $\mathbf{x}_i$ and $\mathbf{x}_j$ should be in different classes, our goal is to classify $\mathcal{X}$ into k classes such that not only the constraints are satisfied, but also those unlabeled objects similar to two must-link objects respectively are classified into the same class and those similar to two cannot-link objects respectively are classified into different classes.

To better illustrate our purpose, let us consider the classification task on a toy data set shown in Fig. 1(a). Although this data set consists of three separate groups (denoted by different colors and symbols in Fig. 1(a)), it has only two classes (Fig. 1(b)). We argue that the must-link constraint asks merging the outer circle and the inner circle into one class, instead of just merging the two must-link objects; and the cannot-link constraint asks for keeping the middle circle and the outer circles into different classes, not just keeping the two cannot-link objects into different classes. Consequently, the desired classification result is the one shown in Fig. 1(b). It is such a global implication that we interpret the pairwise constraints.

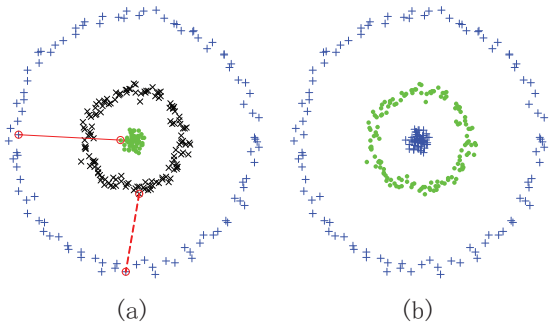


Figure 1. Classification on Three-Circle. (a) A data set with one must-link constraint and one cannot-link constraint, denoted by the solid and the dashed red lines, respectively. (b) Ideal classification (two classes) we hope to obtain where different classes are denoted by different colors and symbols.

From this simple example, we can see that the cluster assumption is still valid, i.e., nearby objects tend to be in the same class and objects on the same manifold structure also tend to be in the same class. However, this assumption does not concern those objects that are not close to each other and do not share the same manifold structure. We argue that the classification for such objects should accord with the input pairwise constraints. For example, any two objects on the outer and inner circles in Fig. 1(a) should be in the same class because they respectively share the same manifold structures with the two must-link objects, and any two objects on the outer and middle circles should be in different classes because they respectively share the same manifold structures with the two cannot-link objects. We refer to this assumption as the *pairwise constraint assumption*.

In this paper, we seek to implement both the cluster assumption and the pairwise constraint assumption in a unified framework. A dilemma is that one may specify nearby objects or objects sharing the same manifold structure to be cannot-link. In this case, we choose to respect the pairwise constraint assumption first and then the cluster assumption, considering that the prior pairwise constraints are from reliable knowledge. This

is true in most practical applications.

3. Pairwise Constraint Propagation

3.1. A General Framework

As mentioned in the last section, our goal is to propagate the given pairwise constraints to the whole data set in a global implication for classification. Intuitively, it is hard to implement our idea in the input space. Therefore, we seek a mapping (usually non-linear) to map the objects to a new and possibly higher-dimensional space such that the objects are reshaped in this way: two must-link objects become close while two cannot-link objects become far apart; objects respectively similar to two must-link objects also become close while objects respectively similar to two cannot-link objects become far apart.

Let $\boldsymbol{\phi}$ be a mapping from $\mathcal{X}$ to some space $\mathcal{F}$,

$$\mathbf{x}_i \in \mathcal{X} \mapsto \boldsymbol{\phi}(\mathbf{x}_i) \in \mathcal{F}. \tag{1}$$

The above analysis motivates us to consider the following optimization framework:

$$\min_{\boldsymbol{\phi}} : \ \mathcal{S}(\boldsymbol{\phi}) \tag{2}$$

$$\text{s.t.} : \ \|\boldsymbol{\phi}(\mathbf{x}_i) - \boldsymbol{\phi}(\mathbf{x}_j)\|_{\mathcal{F}} < \varepsilon, \ \forall(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{M}, \tag{3}$$

$$\|\boldsymbol{\phi}(\mathbf{x}_i) - \boldsymbol{\phi}(\mathbf{x}_j)\|_{\mathcal{F}} > \delta, \ \forall(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{C}, \tag{4}$$

where $\mathcal{S}(\boldsymbol{\phi})$ is a smoothness measure for $\boldsymbol{\phi}$ such that the smaller is $\mathcal{S}(\boldsymbol{\phi})$, the smoother is $\boldsymbol{\phi}$; ε is a small positive number; δ is a large positive number; $\|\cdot\|_{\mathcal{F}}$ is a distance metric in $\mathcal{F}$; $\mathcal{M}$ is the set of the must-link constraints; $\mathcal{C}$ is the set of cannot-link constraints. The inequality constraints (3) and (4) require $\boldsymbol{\phi}$ to map two must-link objects to be close and two cannot-link objects far apart. By enforcing the smoothness on $\boldsymbol{\phi}$ (minimizing the objective (2)), we actually require $\boldsymbol{\phi}$ to map any two objects respectively similar to two must-link objects to be close and map any two objects respectively similar to two cannot-link objects far apart. Hopefully, after the mapping, each class becomes relatively compact and different classes become far apart. Once such a mapping is derived, the classification task can be done much easier.

This optimization framework is quite general and the details have to be developed. We propose a unit hypersphere model to substantialize it in Section 3.2, and then solve the resulting optimization problem in Section 3.3.

3.2. The Unit Hypersphere Model

Recall that our goal is to find a smooth mapping that maps two must-link objects close and two cannot-link objects far apart. To this end, we consider it better to put the images of all the objects under a uniform scale. The unit hypersphere in $\mathcal{F}$ is a good choice because there is a natural way to impose the pairwise constraints on it. Our key idea is to map all the objects onto the unit hypersphere in $\mathcal{F}$, where two must-link objects are mapped to the same point and two cannot-link objects to be orthogonal. Mathematically, we require $\boldsymbol{\phi}$ to satisfy

$$< \boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}_i) >_{\mathcal{F}} = 1, \ i = 1, 2, ..., n, \tag{5}$$

$$< \boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}_j) >_{\mathcal{F}} = 1, \ \forall(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{M}, \tag{6}$$

$$< \boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}_j) >_{\mathcal{F}} = 0, \ \forall(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{C}, \tag{7}$$

where $< \cdot, \cdot >_{\mathcal{F}}$ denotes the dot product in $\mathcal{F}$.

Next we impose smoothness on $\boldsymbol{\phi}$ using the spectral graph theory where the graph Laplacian plays an essential role (Chung, 1997). Let $G = (V, W)$ be an undirected, weighted graph with the node set $V = \mathcal{X}$ and the weight matrix $W = [w_{ij}]_{n \times n}$, where w_{ij} is the weight on the edge connecting nodes $\mathbf{x}_i$ and $\mathbf{x}_j$, denoting how similar they are. W is commonly assumed to be symmetric and non-negative. The graph Laplacian L of G is defined as $L = D - W$, where $D = [d_{ij}]_{n \times n}$ is a diagonal matrix with $d_{ii} = \sum_j w_{ij}$. The normalized graph Laplacian $\bar{L}$ of G is defined as

$$\bar{L} = D^{-1/2} L D^{-1/2} = I - D^{-1/2} W D^{-1/2}, \tag{8}$$

where I is the identity matrix. W is also called the affinity matrix, and $\bar{W} = D^{-1/2} W D^{-1/2}$ the normalized affinity matrix. $\bar{L}$ is symmetric and positive semidefinite, with eigenvalues in the interval $[0, 2]$ (Chung, 1997).

Following the idea of regularization in spectral graph theory (e.g., see (Zhou et al., 2004)), we define the smoothness measure $\mathcal{S}(\cdot)$ by

$$\mathcal{S}(\boldsymbol{\phi}) = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} \| \frac{\boldsymbol{\phi}(\mathbf{x}_i)}{\sqrt{d_{ii}}} - \frac{\boldsymbol{\phi}(\mathbf{x}_j)}{\sqrt{d_{jj}}} \|_{\mathcal{F}}^2, \tag{9}$$

where $\boldsymbol{\phi}(\mathbf{x}_i) \in \mathcal{F}$, and $\|\cdot\|_{\mathcal{F}}$ is a distance metric in $\mathcal{F}$. Note that $\mathcal{F}$ is possibly an infinite-dimensional space. By this definition, we can see that $\mathcal{S}(\boldsymbol{\phi}) \geq 0$ since W is non-negative, and the value $\mathcal{S}(\boldsymbol{\phi})$ penalizes the large change of the mapping $\boldsymbol{\phi}$ between two nodes linked with a large weight. In other words, minimizing $\mathcal{S}(\cdot)$ encourages the smoothness of a mapping over the data graph. Next we rewrite $\mathcal{S}(\boldsymbol{\phi})$ in matrix form.

Let $k_{ij} =< \boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}_j) >_{\mathcal{F}}$. Then the matrix $K = [k_{ij}]_{n \times n}$ is symmetric and positive semidefinite, denoted by $K \succeq 0$, and thus can be thought as a kernel

over $\mathcal{X}$ (Smola & Kondor, 2003). From (9), we have

$$\begin{aligned}\mathcal{S}(\boldsymbol{\phi}) &= \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}\left(\frac{1}{d_{ii}}k_{ii} + \frac{1}{d_{jj}}k_{jj} - 2\frac{1}{\sqrt{d_{ii}d_{jj}}}k_{ij}\right) \\ &= \sum_{i=1}^{n} k_{ii} - \sum_{i,j=1}^{n}\frac{w_{ij}}{\sqrt{d_{ii}d_{jj}}}k_{ij} \quad (10)\\ &= I \bullet K - (D^{-1/2}WD^{-1/2}) \bullet K \quad (11)\\ &= (I - D^{-1/2}WD^{-1/2}) \bullet K = \bar{L} \bullet K, \quad (12)\end{aligned}$$

where $\bullet$ denotes the dot product between two matrices, defined as $A \bullet B = \sum_{i=1}^{n}\sum_{j=1}^{m} a_{ij}b_{ij}$, for $A = [a_{ij}]_{n\times m}$ and $B = [b_{ij}]_{n\times m}$.

3.3. Learning a Kernel Matrix

With the above analysis (5)–(7), and (12), we have arrived at the following optimization problem:

$$\begin{aligned}\min_{K}: \ & \bar{L} \bullet K && (13)\\ \text{s.t.}: \ & k_{ii} = 1, \ i = 1, 2, ..., n, && (14)\\ & k_{ij} = 1, \ \forall(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{M}, && (15)\\ & k_{ij} = 0, \ \forall(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{C}, && (16)\\ & K \succeq 0, && (17)\end{aligned}$$

which can be recognized as a semidefinite programming (SDP) problem (Boyd & Vandenberghe, 2004). This problem is convex and thus the global optimal solution is guaranteed. To solve this problem, we can use the highly optimized software packages, such as SeDuMi (Sturm, 1999) and CSDP (Borchers, 1999).

We can also express the above SDP problem in a more familiar matrix form. Let E_{ij} be a $n \times n$ matrix consisting of all 0's except the $(i, j)^{th}$ entry being 1. Then the above SDP problem becomes

$$\begin{aligned}\min_{K}: \ & \bar{L} \bullet K && (18)\\ \text{s.t.}: \ & E_{ii} \bullet K = 1, \ i = 1, 2, ..., n, && (19)\\ & E_{ij} \bullet K = 1, \ \forall(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{M}, && (20)\\ & E_{ij} \bullet K = 0, \ \forall(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{C}, && (21)\\ & K \succeq 0. && (22)\end{aligned}$$

It should be noted that we have transformed the problem of learning a mapping $\boldsymbol{\phi}$ stated in (2)–(4) into the problem of learning a kernel matrix K such that $\boldsymbol{\phi}$ is the feature mapping induced by K. The kernel trick (Schölkopf & Smola, 2002) indicates that we can implicitly derive $\boldsymbol{\phi}$ by explicitly pursuing K. Note that the kernel matrix K captures the distribution of the point set $\{\boldsymbol{\phi}(\mathbf{x}_i)\}_{i=1}^{n}$ in the feature space. The equality constraints (14) constrain $\boldsymbol{\phi}(\mathbf{x}_i)$'s to be on the unit hypersphere, the inequality constraints (15) and (16) force $\boldsymbol{\phi}(\mathbf{x}_i) = \boldsymbol{\phi}(\mathbf{x}_j)$ if $\mathbf{x}_i$ and $\mathbf{x}_j$ are must-link and $\boldsymbol{\phi}(\mathbf{x}_i)$ and $\boldsymbol{\phi}(\mathbf{x}_j)$ to be orthogonal if $\mathbf{x}_i$ and $\mathbf{x}_j$ are cannot-link. By minimizing the objective function (13), which is equivalent to enforcing smoothness on $\boldsymbol{\phi}$, $\boldsymbol{\phi}(\mathbf{x}_i)$ and $\boldsymbol{\phi}(\mathbf{x}_j)$ will move close to each other if $\mathbf{x}_i$ and $\mathbf{x}_j$ are similar (lie on the same group or manifold). This process will continue until a global stable state is achieved (the objective function is minimized and the constraints are satisfied). We call this process the *pairwise constraint propagation*. It is expected that after the propagation, each class becomes compact and different classes become far apart (being nearly orthogonal on the unit hypersphere). This phenomenon is also observed by our experiments (see Section 5.2). The idea of reshaping the data in a high-dimensional space by propagating the spatial information among objects is previously appeared in our recent work (Li et al., 2007) where the problem of clustering highly noisy data is addressed.

3.4. Classification

Let K^* be the kernel matrix obtained by solving the SDP problem stated in (18)–(22). The final step of our approach is to obtain k classes from K^*. We apply the kernel K-means algorithm (Shawe-Taylor & Cristianini, 2004) to K^* to form k classes.

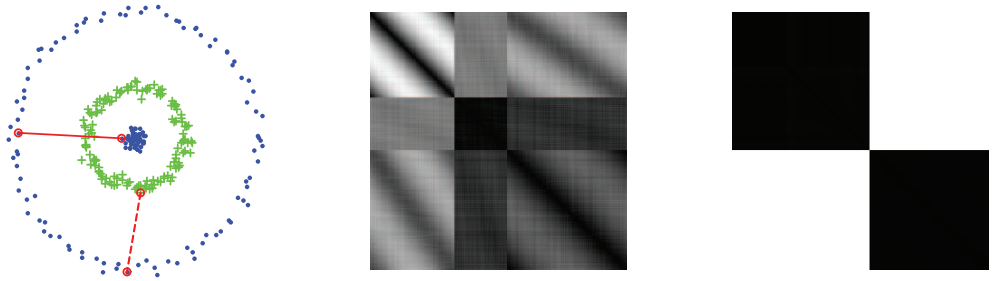
4. The Algorithm

Based on the previous analysis, we develop a semi-supervised classification algorithm listed in Algorithm 1, which we called the *Pairwise Constraint Propagation* (PCP). The scale factor σ in Step 1 needs to be tuned, which is discussed in Section 5.1.

Algorithm 1 Pairwise Constraint Propagation

Input: A data set of n objects $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$, the set of must-link constraints $\mathcal{M} = \{(\mathbf{x}_i, \mathbf{x}_j)\}$, the set of cannot-link constraints $\mathcal{C} = \{(\mathbf{x}_i, \mathbf{x}_j)\}$, and the number of classes k.
Output: The class labels of the objects in $\mathcal{X}$.

1. Form the affinity matrix $W = [w_{ij}]_{n\times n}$ with $w_{ij} = \exp(-d^2(\mathbf{x}_i, \mathbf{x}_j)/2\sigma^2)$ if $i \neq j$ and $w_{ii} = 0$.
2. Form the normalized graph Laplacian $\bar{L} = I - D^{-1/2}WD^{-1/2}$, where $D = \text{diag}(d_{ii})$ is the diagonal matrix with $d_{ii} = \sum_{j=1}^{n} w_{ij}$.
3. Obtain the kernel matrix K^* by solving the SDP problem stated in (18)–(22).
4. Form k classes by applying the kernel K-means to K^*.



(a) (b) (c)

Figure 2. (a) Three-Circle with classes denoted by different colors and symbols. The solid red line denotes a must-link constraint and the dashed red line denotes a cannot-link constraint. (b) & (c) Distance matrices for Three-Circle in the input space and in the feature space, respectively, where, for illustration purpose, the data are arranged such that points within a class appear consecutively. The darker is a pixel, the smaller is the distance the pixel represents.

5. Experimental Results

In this section, we evaluate the proposed algorithm PCP on a number of synthetic and real data sets. By comparison, the results of two notable and most related algorithms, Kamvar et al.'s spectral learning algorithm (SL) (Kamvar et al., 2003) and Kulis et al's semi-supervised kernel K-means algorithm (SSKK) (Kulis et al., 2005), are also presented. Note that most semi-supervised classification algorithms cannot be directly applied to the tasks of classification from pairwise constraints we consider here, because they perform classification from labeled and unlabeled data and cannot be readily generalized to address classification from pairwise constraints and unlabeled data.

In order to evaluate these algorithms, we compare the results with available ground-truth data labels, and employ the *Normalized Mutual Information* (NMI) as the performance measure (Strehl & Ghosh, 2003). For two random variables $\mathbf{X}$ and $\mathbf{Y}$, the NMI is defined as:

$$\mathrm{NMI}(\mathbf{X},\mathbf{Y}) = \frac{I(\mathbf{X},\mathbf{Y})}{\sqrt{H(\mathbf{X})H(\mathbf{Y})}}, \tag{23}$$

where $I(\mathbf{X},\mathbf{Y})$ is the mutual information between $\mathbf{X}$ and $\mathbf{Y}$, and $H(\mathbf{X})$ and $H(\mathbf{Y})$ are the entropies of $\mathbf{X}$ and $\mathbf{Y}$, respectively. Note that $0 \le \mathrm{NMI} \le 1$, and $\mathrm{NMI} = 1$ when a result is the same as the ground-truth. The larger is the NMI, the better is a result.

To evaluate the algorithms under different settings of pairwise constraints, we generate a varying number of pairwise constraints randomly for each data set. For a data set of k classes, we randomly generate j must-link constraints for each class, and j cannot-link constraints for every two classes, giving total $j \times (k + k(k-1)/2)$ constraints for each j, where j ranges from 1 to 10. The averaged NMI is reported for each number of pairwise constraints over 20 different realizations. Since all the three algorithms employ the K-means or kernel K-means in the final step, for each experiment we run the K-means or kernel K-means 20 times with random initializations, and report the averaged result.

5.1. Parameter Selection

The three algorithms are all graph-based and thus the inputs are assumed to be graphs. We use the weighted graphs for all the algorithms, where the similarity matrix $W = [w_{ij}]$ is given by

$$w_{ij} = \begin{cases} e^{-\|\mathbf{x}_i - \mathbf{x}_j\|^2/2\sigma^2} & i \neq j \\ 0 & i = j \end{cases}. \tag{24}$$

The most suitable scale factor σ is found over the set $S(0.1r, r, 5) \bigcup S(r, 10r, 5)$, where $S(r_1, r_2, m)$ denotes the set of m linearly equally spaced numbers between r_1 and r_1, and r denotes the averaged distance from each node to its 10-th nearest neighbor.

We use the SDP solver CSDP 6.0.1[1] (Borchers, 1999) to solve the SDP problem in the proposed PCP. For SSKK, we use its normalized cut version since it performs best in the experiments given in (Kulis et al., 2005). The constraint penalty in SSKK is set to $n/(kc)$, as suggested in (Kulis et al., 2005), where n is the number of objects, k is the number of classes, and c is the total number of pairwise constraints. All the algorithms are implemented in MATLAB 7.6, running on a 3.4 GHz, 2GB RAM Pentium IV PC.

5.2. A Toy Example

In this subsection, we illustrate the proposed PCP using a toy example. We mainly study its capability of propagating pairwise constraints to the whole data

[1] https://projects.coin-or.org/Csdp/.

Table 1. Description of the four sensory data sets from UCI.

Data	Iris	Wine	Ionosphere	Soybean
Number of objects	150	178	351	47
Dimension	4	13	34	35
Number of classes	3	3	2	4

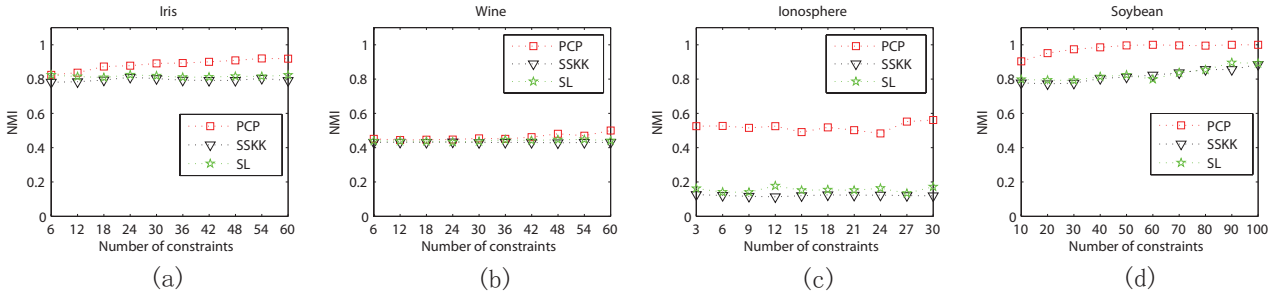


Figure 3. Classification results on the four sensory data sets: NMI vs. Number of constraints. (a) Results on Iris. (b) Results on Wine. (c) Results on Ionosphere. (d) Results on Soybean.

set. Specifically, we want to see how PCP reshapes the data in the feature space according to the original data structure and the given pairwise constraints. The classification task on the Three-Circle data is shown in Fig. 2(a), where the ground-truth classes are denoted by different colors and symbols, and one must-link (solid red line) and one cannot-link (dashed red line) constraints are also provided. At first glance, Three-Circle is composed of a mixture of curve-like and Gaussian-like groups. A more detailed observation is that there is one class composed of separate groups.

The distance matrices for Three-Circle in the input space and in the feature space are shown in Figs. 2(b) and (c), where the data are ordered such that all the objects in the outer circle appear first, all the objects in the inner circle appear second, and all the objects in the middle circle appear finally. Note that this arrangement does not affect the classification results but only for better illustration of the distance matrices. We can see that the distance matrix in the feature space, compared to the one in the input space, exhibits a clear block structure, meaning that each class becomes highly compact (although in the input space one of the two classes consists of two well-separated groups) and the two classes become far apart. Our computations show that the distance between any two points in the feature space in different classes falls in $[\sqrt{2} - 1.9262 \times 10^{-5}, \sqrt{2} + 1.3214 \times 10^{-6}]$, implying that the two classes are nearly $\sqrt{2}$ from each other, which comes from the requirement that two cannot-link objects are mapped to be orthogonal on the unit hypersphere.

5.3. On Sensory Data

Four sensory data sets from the UCI Machine Learning Repository[2] are used for testing in this experiment. The data sets are described in Table 1. These four data sets are widely used for evaluation of the classification and clustering methods in the machine learning community.

The results are shown in Fig. 3, from which two observations can be drawn. First, PCP performs better than SSKK and SL on all the four data sets under different settings of pairwise constraints, especially on the Ionosphere data. Second, as the number of constraints grows, the performances of all the algorithms increase accordingly on Soybean, but vary little on Wine and Ionosphere. On Iris, as the number of constraints grows the performance of PCP improves accordingly but those of SSKK and SL are almost unchanged.

5.4. On Imagery Data

In this subsection, we test the algorithms on three image databases USPS, MNIST, and CMU PIE (Pose, Illumination, and Expression). Both USPS and MNIST consist of images of handwritten digits with significantly different fashion and of sizes 16×16 and 28×28. The CMU PIE contains 41,368 images of 68 people, each person with 13 different poses, 43 different illumination conditions, and 4 different expressions. From these databases, we draw four data sets, which are described in Table 2. The USPS0123 and MNIST0123 are drawn respectively from USPS and MNIST, and PIE-10-20 and PIE-22-23 are drawn from CMU PIE.

[2] http://archive.ics.uci.edu/ml.

Table 2. Description of the four imagery data sets.

Data	USPS0123	MNIST0123	PIE-10-20	PIE-22-23
Number of objects	400	400	340	340
Dimension	256	784	1024	1024
Number of classes	4	4	2	2

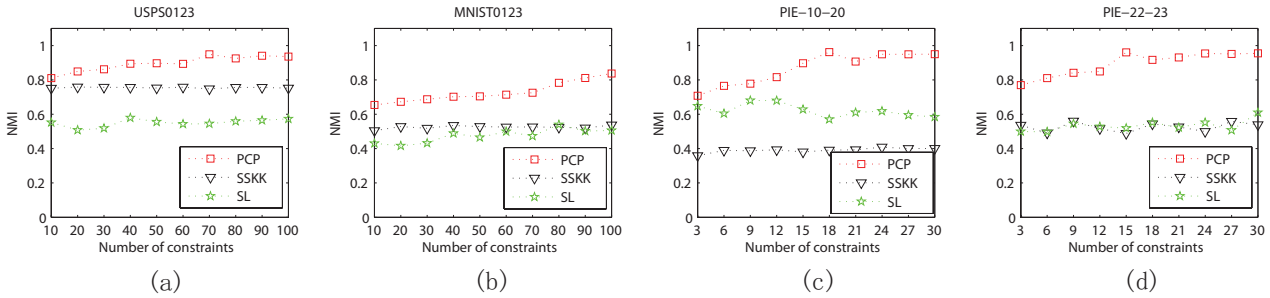


Figure 4. Classification results on the four imagery data sets: NMI vs. Number of constraints. (a) Results on USPS0123. (b) Results on MNIST0123. (c) Results on PIE-10-20. (d) Results on PIE-22-23.

USPS0123 (MNIST0123) consists of digits 0 to 3, with first 100 instances from each class. PIE-10-20 (PIE-22-23) contains five near frontal poses (C05, C07, C09, C27, C29) of two individuals indexed as 10 and 20 (22 and 23) under different illuminations and expressions. Original images in PIE-10-20 and PIE-22-23 are manually aligned (two eyes are aligned at the fixed positions), cropped, and then down-sampled to 32×32. Each image is represented by a vector of size equal to the product of its width and height.

The results are shown in Fig. 4, from which we can see that the proposed PCP consistently and significantly outperforms SSKK and SL on all the four data sets under different settings of pairwise constraints. As the number of constraints grows, the performance of PCP improves more significantly than those of SSKK and SL.

We also look at the computational costs of different algorithms. For example, for each run on USPS0123 (of size 400) with 100 pairwise constraints, PCP takes about 17 seconds while both SSKK and SL take less than 0.5 second. PCP does take more execution time than SSKK and SL since it involves solving for a kernel matrix with SDP, while either SSKK or SL uses predefined kernel matrix. The main computational cost in PCP is in solving the SDP problem.

6. Conclusions

A semi-supervised classification approach, *Pairwise Constraint Propagation* (PCP), for learning from pairwise constraints and unlabeled data is proposed. PCP seeks a smooth mapping to map the data onto a unit hypersphere, where any two must-link objects are mapped to the same point and any two cannot-link objects are mapped to be orthogonal. Consequently, PCP simultaneously implements the cluster assumption and the pairwise constraint assumption stated in Section 2. PCP implicitly derives such a mapping by explicitly finding a kernel matrix via semidefinite programming. In contrast to label propagation in traditional semi-supervised learning, PCP can effectively propagate pairwise constraints to the whole data set. Experimental results on a variety of synthetic and real data sets have demonstrated the superiority of PCP.

Note that PCP falls into semi-supervised learning since it performs learning from both constrained and unconstrained data. Most previous metric learning methods, however, belong to supervised learning. PCP always keeps every two must-link objects close and every two cannot-link objects far apart. Therefore it essentially addresses hard constrained classification.

Although extensive experiments have confirmed the effectiveness of the PCP algorithm, there are several issues worthy to be further investigated in future work. One issue is to accelerate PCP where solving the associated SDP problem is the bottleneck. Another issue is to handle noisy constraints effectively.

Acknowledgments

We would like to thank the anonymous reviewers for their valuable comments. The work described in this paper was fully supported by a grant from the Research Grants Council of the Hong Kong SAR, China (Project No. CUHK 414306).

References

Bar-Hillel, A., Hertz, T., Shental, N., & Weinshall, D. (2003). Learning distance functions using equivalence relations. *ICML* (pp. 11–18).

Basu, S., Bilenko, M., & Mooney, R. (2004). A probabilistic framework for semi-supervised clustering. *SIGKDD* (pp. 59–68).

Belkin, M., Niyogi, P., & Sindhwani, V. (2006). Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, *7*, 2399–2434.

Bilenko, M., Basu, S., & Mooney, R. (2004). Integrating constraints and metric learning in semi-supervised clustering. *ICML*.

Borchers, B. (1999). CSDP, a C library for semidefinite programming. *Optimization Methods & Software*, *11-2*, 613–623.

Boyd, S., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge University Press.

Chapelle, O., Schölkopf, B., & Zien, A. (2006). *Semi-supervised learning*. MIT Press.

Chapelle, O., & Zien, A. (2005). Semi-supervised classification by low density separation. *AISTATS*.

Chung, F. (1997). *Spectral graph theory*. American Mathematical Society.

Globerson, A., & Roweis, S. (2006). Metric learning by collapsing classes. *Advances in Neural Information Processing Systems* (pp. 451–458).

Goldberg, A., Zhu, X., & Wright, S. (2007). Dissimilarity in graph-based semisupervised classification. *AISTATS*.

Hoi, S., Jin, R., & Lyu, M. (2007). Learning nonparametric kernel matrices from pairwise constraints. *ICML* (pp. 361–368).

Kamvar, S., Klein, D., & Manning, C. (2003). Spectral learning. *IJCAI* (pp. 561–566).

Klein, D., Kamvar, S., & Manning, C. (2002). From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering. *ICML* (pp. 307–314).

Kulis, B., Basu, S., Dhillon, I., & Mooney, R. (2005). Semi-supervised graph clustering: A kernel approach. *ICML* (pp. 457–464).

Li, Z., Liu, J., Chen, S., & Tang, X. (2007). Noise robust spectral clustering. *ICCV*.

Schölkopf, B., & Smola, A. (2002). *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT Press.

Shawe-Taylor, J., & Cristianini, N. (2004). *Kernel methods for pattern analysis*. Cambridge University Press.

Smola, A., & Kondor, R. (2003). Kernels and regularization on graphs. *COLT*.

Strehl, A., & Ghosh, J. (2003). Cluster ensembles — a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, *3*, 583–617.

Sturm, J. F. (1999). Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods & Software*, *11-2*, 625–653.

Szummer, M., Jaakkola, T., & Cambridge, M. (2002). Partially labeled classification with markov random walks. *Advances in Neural Information Processing Systems*.

Tong, W., & Jin, R. (2007). Semi-supervised learning by mixed label propagation. *AAAI*.

Wagstaff, K., & Cardie, C. (2000). Clustering with instance-level constraints. *ICML* (pp. 1103–1110).

Wagstaff, K., Cardie, C., Rogers, S., & Schroedl, S. (2001). Constrained k-means clustering with background knowledge. *ICML* (pp. 577–584).

Xing, E., Ng, A., Jordan, M., & Russell, S. (2003). Distance metric learning, with application to clustering with side-information. *Advances in Neural Information Processing Systems* (pp. 505–512).

Zhang, T., & Ando, R. (2006). Analysis of spectral kernel design based semi-supervised learning. *Advances in Neural Information Processing Systems* (pp. 1601–1608).

Zhou, D., Bousquet, O., Lal, T. N., Weston, J., & Schölkopf, B. (2004). Learning with local and global consistency. *Advances in Neural Information Processing Systems*.

Zhu, X. (2005). *Semi-supervised learning literature survey* (Technical Report 1530). Computer Sciences, University of Wisconsin-Madison.

Zhu, X., Ghahramani, Z., & Lafferty, J. Semi-supervised learning using gaussian fields and harmonic functions. *ICML* (pp. 912–919).