Local similarity learning for pairwise constraint propagation

Zhenyong Fu · Zhiwu Lu · Horace H. S. Ip ·
Hongtao Lu · Yunyun Wang

Published online: 16 January 2014

© The Author(s) 2014. This article is published with open access at Springerlink.com

Abstract Pairwise constraint propagation studies the problem of propagating the scarce pairwise constraints across the entire dataset. Effective propagation algorithms have previously been designed based on the graph-based semi-supervised learning framework. Therefore, these previous constraint propagation methods rely critically on a good similarity measure over the data points. Improper or noisy similarity measurements may dramatically degrade the performance of the constraint propagation algorithms. In this paper, we make attempt to exploit the available pairwise constraints to learn a new set of similarities, which are consistent with the supervisory information in the pairwise constraints, before propagating these initial constraints. Our method is a local learning algorithm. More specifically, we compute the similarities at each data point through simultaneously minimizing the local reconstruction error and local constraint error. The proposed method has been tested in the constrained clustering tasks on eight real-life datasets and then shown to achieve significant improvements with respect to the state of the arts.

Z. Fu (✉) · Y. Wang
College of Computer, Nanjing University of Posts and Telecommunications,
Nanjing, Jiangsu 210003, China
e-mail: fuzy@njupt.edu.cn

Y. Wang
e-mail: wangyunyun@njupt.edu.cn

Z. Lu
Department of Computer Science, School of Information, Renmin University of China,
Beijing, China
e-mail: luzhiwu@ruc.edu.cn

H. H. S. Ip
Department of Computer Science, City University of Hong Kong, Hong Kong, Hong Kong
e-mail: cship@cityu.edu.hk

H. Lu
Department of Computer Science and Engineering, Shanghai Jiao Tong University,
Shanghai, China
e-mail: htlu@sjtu.edu.cn

Keywords Constrained clustering · Pairwise constraint propagation · Similarity learning · Semi-supervised learning

1 Introduction

Prior knowledge on whether two objects belong to the same cluster or not are expressed respectively in terms of must-link constraints and cannot-link constraints, the pairwise constraints [2, 19]. Generally, it is hard to infer instance labels only from pairwise constraints, especially for multi-class data. This means that pairwise constraints are weaker and thus more general than the explicit labels of data.

Pairwise constraints have been widely used in the context of clustering with side information [1, 7, 8, 23], where it has been shown that the presence of appropriate pairwise constraints can often improve the performance. For instance, one way to deal with pairwise constraints is to trivially set the similarities between the constrained data to 1 and 0 for must-link and cannot-link constraints, respectively [7]. In general, while it is possible to infer pairwise constraints from domain knowledge or user feedback, in practice, the availability of such pairwise constraints is scarce. Thus, it can not achieve much more performance improvement to only adjust the similarities between constrained data.

One more effective approach is to propagate such scarce pairwise constraints across all the data points for fully utilizing the information inherent in the available pairwise constraints. The problem of pairwise constraint propagation differs from that of label propagation and is more challenging in the following aspects: (a) unlike class labels for data, pairwise constraints in general do not provide explicit class information; (b) it is in general not possible to infer class label directly from the pairwise constraints which simply state whether a pair of data belong to the same class or not; and (c) for a dataset of size n , there are potentially $O(n^2)$ pairwise constraints that can be inferred through constraint propagation, while there are only $O(n)$ class labels that need to be inferred for the data in label propagation.

Due to the scarcity of the available pairwise constraints, the problem of pairwise constraint propagation can be viewed as a kind of semi-supervised learning (SSL) problem. People wish to resort the massive unconstrained data to help propagate the scarce pairwise constraints, in other words, to solve the challenging problem of pairwise constraint propagation based on the traditional graph-based SSL framework. For instance, Lu and Ip propose an exhaustive and efficient constraint propagation method (E^2CP), which decomposes the constraint propagation problem as a series of independent label propagation (SSL) subproblems [14].

The basic assumption behind most of the graph-based algorithms is the so-called *similar* assumption, which states that two objects are likely to have the same properties if they are similar with each other. Therefore, to the graph-based methods, it is very important to firstly construct a similarity graph, which is used to reflect the similar/dissimilar relationships between instances. For instance, the E^2CP needs the constructed graph to tell which two data points are similar and which two are not similar. Similar objects will share the same propagated constraint settings with respect to other objects, while dissimilar objects will have different propagated constraint settings. Thus, the performance of constraint propagation algorithms based on SSL framework heavily depends on the similarity measurements between instances.

In the traditional graph-based methods, the choice to the computation manner of the similarity measurements is usually trivial. For instance, it is common to use the gaussian

function to compute the similarity between two data points. One better manner is to apply the domain knowledge to compute the similarity, such as the spatial pyramid matching (SPM) kernel [9] and spatial markov kernel (SMK) [13, 15] in the computer vision area. However, the similar/dissimilar relationships between data points reflected by the computed similarity measurements may not necessarily correspond to the real categories of the data. More specifically, two data points with the higher similarity may not have the same class label, i.e., may not have the must-link constraint relationship, and vice versa. These practical issues may all be due to that the traditional similarity measurement computation mainly works in a completely unsupervised setting and does not take any prior knowledge into consideration. The current methods use the graph constructed through the unsupervised manner to propagate the supervised information about the category relationships between objects, which is incorporated in the pairwise constraints. Since the graph construction manner in the traditional graph-based semi-supervised learning is unsupervised, the performance of constraint propagation methods based on the SSL framework, such as E²CP, may suffer from the improper similarity measurements.

To overcome the above issue, in this paper, we make attempt to exploit the available pairwise constraints to learn better similarity measurements between objects before propagating these initial pairwise constraints. With the help of the pairwise constraints, the graph construction process will be no longer unsupervised, but supervised. Therefore, in some extent, the learnt similarity measurements will be consistent with the category information reflected by the pairwise constraints. At the same time, the proposed similarity learning method also takes the original similarity measurements into account because the supervisory information provided by the initial pairwise constraints is still scarce. The proposed approach is a local similarity learning method, that is, we compute a set of new similarity measurements for each object in the dataset. More specifically, at each object, we simultaneously minimize the *local* reconstruction error and the *local* constraint error. Each similarity learning subproblem at each data point can be formulated as a convex quadratic programming problem, which can be solved efficiently. Once all of the similarity learning subproblems are computed, we can form a new similarity graph with the new similarity measurements.

The learnt similarity measurements are consistent with the real category relationships of the data, therefore, the following constraint propagation process can benefit from these new similarity measurements. To evaluate the effectiveness of the proposed method, we select eight real-life datasets for constrained clustering tasks. The experimental results have shown that the proposed method can achieve significant improvements with respect to the state of the arts.

The remainder of this paper is organized as follows. Section 2 analyzes related works. In Section 3, we present the proposed similarity learning method with the available pairwise constraints and then the details of pairwise constraint propagation using the learnt similarity graph. In Section 4, our method is evaluated on eight real-life datasets. Finally, Section 5 gives the conclusions drawn from the results.

2 Related works

Pairwise constraints have been widely used in the context of clustering with side information [1, 7, 8, 23], where it has been shown that the presence of appropriate pairwise constraints can often improve the performance. One way to deal with pairwise constraints is to trivially set the similarities between the constrained data to 1 and 0 for must-link and cannot-link

constraints, respectively [7]. In [22], Wang and Davidson proposed a flexible constrained spectral clustering algorithm, which can be solved in polynomial time and also allows soft constraints. However, their method is only valid for two-class problems, and is not applicable to multi-class problems. Another work of Wang and Davidson in [21] suffers from the same limitation.

In [10], a spectral kernel learning framework is proposed for the constrained clustering task (CCSKL). Similar to our work, this method learns a new similarity measurements of the data with the pairwise constraints. However, this method differs from our work in the following aspects: (a) the CCSKL is a global similarity learning method, while our work is a locally learning method; (b) the CCSKL does not propagate the scarce pairwise constraints among the data points, while our work applies the learnt similarity measurements to further propagate the initial scarce pairwise constraints; and (c) the experimental results show that our work significantly outperforms the CCSKL, especially, the CCSKL can not benefit from the increasing number of the constraints, while our work achieves a unanimous and obvious improvement in the performance as the number of initial pairwise constraints increases.

The above methods do not propagate the constraint information from the constrained data to other data. In contrast, in [12], the pairwise constraints are propagated to unconstrained data using the Gaussian process. However, as noted in [12], this method makes certain assumptions for constraint propagation especially with respect to two-class problems. Although an heuristic approach for multi-class problems is also discussed, it incurs a large time cost. In [11], pairwise constraint propagation is formulated as a semidefinite programming (SDP) problem. Although this optimization-based method is not limited to two-class problems, it leads to extremely large time cost for solving SDP and is not particularly scalable to large datasets. In [14], an exhaustive and efficient pairwise constraint propagation method is developed based on semi-supervised learning. Their work provides a tractable framework that shows how pairwise constraints are propagated independently and then accumulated into a conciliatory closed-form solution. However, the method proposed by Lu and Ip only utilizes the original similarity graph predefined on the dataset to make the constraint propagation. This method may suffer from the improper similarity measurements because the original similarity graph does not explore any supervisory information in its construction and thus may be inconsistent with the ground-truth.

3 Learning the similarities for pairwise constraint propagation

In this section, we first give the problem formulation, followed by the details of the proposed similarity learning with the pairwise constraints and the constraint propagation using the learnt similarity graph.

3.1 Problem formulation

Given a dataset of n objects $\mathcal{X} = \{x_1, \dots, x_n\}$ and two sets of pairwise constraints, denoted respectively by $\mathcal{M} = \{(x_i, x_j)\}$ where x_i and x_j should be in the same class and $\mathcal{C} = \{(x_i, x_j)\}$ where x_i and x_j should be in different classes, the goal of pairwise constraint propagation is to propagate these pairwise constraints across the entire dataset.

To propagate the initial pairwise constraints on $\mathcal{X}$, we first represent the two types of pairwise constraints, i.e. the $\mathcal{M}$ and $\mathcal{C}$, with a single matrix $Y = \{Y_{ij}\}_{n \times n}$:

$$Y_{ij} = \begin{cases} +1, & (x_i, x_j) \in \mathcal{M}; \\ -1, & (x_i, x_j) \in \mathcal{C}; \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

Given $\mathcal{X}$ and Y , the problem of pairwise constraint propagation would be to find a matrix $F = \{F_{ij}\}_{n \times n}$ so that for any pair instances $(x_i, x_j) \in \mathcal{X} \times \mathcal{X}$, the element F_{ij} can be used to predict the pairwise constraint relationship between x_i and x_j , i.e., $F_{ij} > 0$ means that x_i and x_j should be must-link while $F_{ij} < 0$ means that x_i and x_j should be cannot-link.

Let $\mathcal{G} = (\mathcal{V}, \mathcal{K})$ be an undirected, weighted graph prior defined on the dataset, i.e., $\mathcal{V} = \mathcal{X}$. The similarity (kernel) matrix $\mathcal{K}$ is defined as $\mathcal{K} = [\kappa_{ij}]_{n \times n}$, where κ_{ij} is the similarity measurement between x_i and x_j on an (inner product) feature space. We can write $\kappa_{ij} = \langle \phi(x_i), \phi(x_j) \rangle$, where ϕ is an (implicit) mapping from $\mathcal{X}$ to the feature space. For modeling the local neighborhood relationships between the data points, we construct a k -nearest neighbor graph. Let $\mathcal{N}_k(x_i)$ represent the k -nearest neighborhood of x_i , then $\kappa_{ij} = 0$ if $x_j \notin \mathcal{N}_k(x_i)$.

For simplicity, we use $Y_i = (Y_{ij}(x_j))$ to denote the initial pairwise constraints between x_i and each $x_j \in \mathcal{N}_k(x_i)$. Obviously, the elements of Y_i come from the i -th column (or row) of Y , which are actually the prior pairwise constraints setting between x_i and the other data points.

3.2 Learning the similarities with pairwise constraints

Instead of using the prior defined similarity graph directly, various graph construction methods have been proposed to apply the neighborhood information of each data point to reconstruct a new similarity graph. For example, the *Linear Neighborhood Propagation* assumes that all of these neighborhoods are linear [20], that is, each data point can be optimally reconstructed using a linear combination of its neighbors on the feature space.

For each data point $x_i \in \mathcal{X}$, we want to determine a new set of similarity measurements, $\mathbf{w}_i$, between x_i and the other data points. For $x_j \notin \mathcal{N}_k(x_i)$, we simply set $\mathbf{w}_i(x_j) = 0$. For $x_j \in \mathcal{N}_k(x_i)$, we can compute $\mathbf{w}_i(x_j)$ through minimizing the following cost function, i.e. the “local reconstruction error” defined on x_i :

$$\epsilon_i^r = \|\phi(x_i) - \sum_{x_j \in \mathcal{N}_k(x_i)} \mathbf{w}_i(x_j) \phi(x_j)\|_2^2. \quad (2)$$

ϵ_i^r measures the linear reconstruction error of x_i with its k -nearest neighbors $\mathcal{N}_k(x_i)$, where $\mathbf{w}_i(x_j)$ is the contribution of $x_j \in \mathcal{N}_k(x_i)$ to the linear reconstruction of x_i . Intuitively, the more similar $x_j \in \mathcal{N}_k(x_i)$ is to x_i , the more correlated these two data points are, and then the larger contribution $\mathbf{w}_i(x_j)$ will be. Thus, we can use $\mathbf{w}_i(x_j)$ as the new similarity measurement between x_i and $x_j \in \mathcal{N}_k(x_i)$. In the following, for simplicity, we use $\mathbf{w}_i$ to denote the $1 \times k$ vector, $\mathbf{w}_i = (\mathbf{w}_i(x_j))_{1 \times k}$, that stores the similarity weights between x_i and $x_j \in \mathcal{N}_k(x_i)$.

Similar to linear neighborhood propagation, we further assume that $\sum_j \mathbf{w}_i(x_j) = 1$ and

$\mathbf{w}_i(x_j) \geq 0$ for $x_j \in \mathcal{N}_k(x_i)$. The objective function defined in (2) can be transformed into the following compact form

$$\begin{aligned}\epsilon_i^r &= \|\phi(x_i) - \sum_{x_j} \mathbf{w}_i(x_j)\phi(x_j)\|_2^2 \\ &= \|\sum_{x_j} \mathbf{w}_i(x_j)[\phi(x_i) - \phi(x_j)]\|_2^2 \\ &= \sum_{x_j} \sum_{x_k} \mathbf{w}_i(x_j)G_i(j, k)\mathbf{w}_i(x_k) \\ &= \mathbf{w}_i^T G_i \mathbf{w}_i,\end{aligned}\quad (3)$$

where G_i is the local Gram matrix around x_i , in which $G_i(j, k) = [\phi(x_i) - \phi(x_j)]^T [\phi(x_i) - \phi(x_k)]$ represents the (j, k) -th entry of G_i . We can directly compute the entries of G_i using the kernel matrix $\mathcal{K}$. That is

$$\begin{aligned}G_i(j, k) &= [\phi(x_i) - \phi(x_j)]^T [\phi(x_i) - \phi(x_k)] \\ &= \langle \phi(x_i), \phi(x_i) \rangle - \langle \phi(x_i), \phi(x_k) \rangle \\ &\quad - \langle \phi(x_j), \phi(x_i) \rangle + \langle \phi(x_j), \phi(x_k) \rangle.\end{aligned}$$

According to the definition of $\mathcal{K}$, we have

$$G_i(j, k) = \kappa_{ii} - \kappa_{ik} - \kappa_{ji} + \kappa_{jk}. \quad (4)$$

However, the above similarity learning process only considers the prior computed similarity measurements, but does not take any supervisory information, e.g. the pairwise constraints, into consideration. As discussed previously, we hope to use the computed $\mathbf{w}_i$ at x_i to represent the learnt similarity measurements between x_i and the other data points. Besides that $\mathbf{w}_i$ should minimize the “local reconstruction error”, $\mathbf{w}_i$ should also be consistent with the initial pairwise constraints. This is a kind of supervised learning inspired by the idea of exploiting the initial pairwise constraint relationships among the data points for similarity learning.

Intuitively, due to each $\mathbf{w}_i(x_j) \in [0, 1]$, if $(x_i, x_j) \in \mathcal{M}$, then $\mathbf{w}_i(x_j)$ should be closer to 1, which means that (x_i, x_j) tends to have the biggest similarity. In contrast, if $(x_i, x_j) \in \mathcal{C}$, then $\mathbf{w}_i(x_j)$ should be closer to 0, which means that (x_i, x_j) tends to have the smallest similarity. Let $H_i = (h_i(x_j))_{1 \times k}$ be the constraint indicator vector defined on $\mathcal{N}_k(x_i)$ where for $x_j \in \mathcal{N}_k(x_i)$, $h_i(x_j) = 1$ if $(x_i, x_j) \in \mathcal{M}$ or $(x_i, x_j) \in \mathcal{C}$, and $h_i(x_j) = 0$ otherwise. The above intuition can be expressed as minimizing the following cost function, i.e. the “local constraint error” defined on x_i :

$$\epsilon_i^c = \|H_i \circ (\mathbf{w}_i - Y_i)\|_2^2, \quad (5)$$

where $\circ$ denotes element-wise product between two vectors.

The above analysis suggests the following optimization problem:

$$\begin{aligned}\min_{\mathbf{w}_i} \quad & \epsilon_i^r + \lambda \epsilon_i^c + \mu \|\mathbf{w}_i\|_2^2 \\ \text{s.t.} \quad & \sum_{x_j} \mathbf{w}_i(x_j) = 1, \mathbf{w}_i \geq 0,\end{aligned}\quad (6)$$

where λ is an optimization parameter and μ is the regularization parameter. In our experiments below, we set $\lambda = 0.1$ and $\mu = 0.1$. Moreover, the objective function in (6) can be reformulated as

$$\begin{aligned} & \mathbf{w}_i^T G_i \mathbf{w}_i + \lambda \|H_i \circ (\mathbf{w}_i - Y_i)\|_2^2 + \mu \|\mathbf{w}_i\|_2^2 \\ &= \frac{1}{2} \mathbf{w}_i^T A \mathbf{w}_i + b^T \mathbf{w}_i + c, \end{aligned} \quad (7)$$

in which

$$A = 2[G_i + \lambda \text{diag}(H_i \circ H_i) + \mu I] \quad (8a)$$

$$b = -2\lambda[\text{diag}(H_i \circ H_i)Y_i] \quad (8b)$$

$$c = \lambda Y_i^T \text{diag}(H_i \circ H_i)Y_i, \quad (8c)$$

where I is the identity matrix and $\text{diag}(H_i \circ H_i)$ is a diagonal matrix with the j -th diagonal element being the j -th element of the vector $H_i \circ H_i$. Therefore, the problem (6) is equivalent to

$$\begin{aligned} \min_{\mathbf{w}_i} \quad & \frac{1}{2} \mathbf{w}_i^T A \mathbf{w}_i + b^T \mathbf{w}_i \\ \text{s.t.} \quad & \sum_{x_j} \mathbf{w}_i(x_j) = 1, \mathbf{w}_i \geq 0, \end{aligned} \quad (9)$$

where the constant term c is dropped.

Our formulation for similarity learning is a standard quadratic programming problem with k variables [3]. In our experiments, we fix $k = 10$, and thus for each $x_i \in \mathcal{X}$ the (9) is a small-scale optimization problem. For a dataset of size n , the time complexity of our similarity learning process is therefore $O(n)$. Also note that A is symmetric and positive definite. Therefore this quadratic programming problem is convex and can be optimally solved efficiently.

In summary, in the similarity learning step, for each $x_i \in \mathcal{X}$, we solve the (9) to compute the similarity measurements about x_i , i.e. the $\mathbf{w}_i$. Once all of the $\mathbf{w}_i$ for each x_i are computed, we will construct a new similarity matrix W by setting

$$W_{ij} = \begin{cases} \mathbf{w}_i(x_j), & x_j \in \mathcal{N}_k(x_i); \\ 0, & \text{otherwise.} \end{cases} \quad (10)$$

In other words, W forms the edge weights of the learnt similarity graph.

3.3 Pairwise constraint propagation with the learnt similarities

In this section, we present the details of pairwise constraint propagation process using our learnt similarities. In fact, the learnt similarity graph can be used to combine with any existing constraint propagation algorithm to propagate the initial pairwise constraints across all the data points. In this paper, we adopt the exhaustive and efficient constraint propagation (E²CP) method described in [14], which decomposes the problem of pairwise constraint propagation into a series of two-class label propagation subproblems and then makes use of semi-supervised learning to solve each propagation subproblem [24, 25].

We set $W_{ii} = 0$ for $1 \leq i \leq n$ to avoid self-reinforcement, and set $W = (W^T + W)/2$ to ensure that W is symmetric. The normalized graph Laplacian $\mathcal{L}$ of the learnt similarity graph is defined as

$$\mathcal{L} = I - D^{-1/2} W D^{-1/2},$$

where $D = [d_{ii}]_{n \times n}$ is a diagonal matrix with $d_{ii} = \sum_j W_{ij}$ [4].

Fig. 1 One example of pairwise constraints indicator matrix as defined in (1)

	x_1	x_2	x_3	x_4	x_5	x_6	x_7
x_1	0	-1	1	0	0	0	1
x_2	-1	0	0	-1	0	0	0
x_3	1	0	0	0	0	-1	0
x_4	0	-1	0	0	0	0	0
x_5	0	0	0	0	0	0	0
x_6	0	0	-1	0	0	0	-1
x_7	1	0	0	0	0	-1	0

A standard way to compute the constraint propagation result F is to minimize the loss between F and Y , and at the same time minimize the regularization of F on $\mathcal{X}$ [14]. As shown in Fig. 1, the E²CP algorithm includes the propagation processes respectively along the vertical and the horizontal direction of the pairwise constraint matrix defined in (1). The vertical propagation results F_v can be computed by solving the following optimization problem:

$$\min_{F_v} \frac{1}{2} \eta \|F_v - Y\|_F^2 + \frac{1}{2} \text{tr}(F_v^T \mathcal{L} F_v),$$

where η is an optimization parameter, $\text{tr}(Z)$ stands for the trace of a matrix Z and $\|\cdot\|_F$ is the Frobenius norm. The closed-form results of the vertical pairwise constraint propagation F_v is:

$$F_v = \eta(\eta I + \mathcal{L})^{-1} Y.$$

In addition, the horizontal propagation results F_h can be computed by solving the following optimization problem:

$$\min_{F_h} \frac{1}{2} \eta \|F_h - Y\|_F^2 + \frac{1}{2} \text{tr}(F_h \mathcal{L} F_h^T),$$

and the results of horizontal pairwise constraint propagation F_h is:

$$F_h = \eta Y(\eta I + \mathcal{L})^{-1}.$$

In [14], Lu and Ip state that if only considering constraint propagation along one direction, i.e. either vertically or horizontally, it is possible that a column (or a row) contains no pairwise constraints, such as the fifth column (row) described in Fig. 1. This problem can be dealt with by performing the pairwise constraint propagation process twice, first along the vertical direction, and then along the horizontal direction. If we use the vertical results F_v as the initial constraints in the horizontal propagation, the final closed-form solution can be obtained:

$$F = \eta^2(\eta I + \mathcal{L})^{-1} Y(\eta I + \mathcal{L})^{-1}. \quad (11)$$

3.4 The LSCP algorithm

The overall procedure of the proposed method is summarized in Algorithm 1, which we call learning similarity for constraint propagation (LSCP).

4 Experimental results

In this section, we evaluate the performances of our LSCP by applying to constrained clustering on eight real-life datasets. In the following, we first describe how to incorporate the constraint propagation results into the clustering process. Then we describe the experimental setup, including the performance measure and the parameter selection. Finally we compare our LSCP with other five methods on the four image datasets and four UCI datasets, respectively.

Algorithm 1 Learning Similarity for Constraint Propagation

Input: A dataset of n objects $\mathcal{X} = \{x_1, x_2, \dots, x_n\}$, the set of must-link constraints $\mathcal{M} = \{(x_u, x_v)\}$, the set of cannot-link constraints $\mathcal{C} = \{(x_u, x_v)\}$.

Output: The result matrix F of constraint propagation.

1. Form the k -NN similarity (kernel) matrix $\mathcal{K} = [\kappa_{ij}]n \times n$.
 2. Generate the initial constraint matrix Y with $\mathcal{M}$ and $\mathcal{C}$ using (1).
 3. Learn the similarity measurements by solving the equation (9) for each x_i , and form the new similarity matrix W using (10).
 4. Form the normalized graph Laplacian $\mathcal{L} = I - D^{-1/2}WD^{-1/2}$, where $D = \text{diag}(d_{ii})$ is the diagonal matrix with $d_{ii} = \sum_{j=1}^n W_{ij}$.
 5. Compute the constraint propagation result F by (11).
-

4.1 Application of constraint propagation to constrained clustering

Once we have obtained the constraint propagation result F , we can consider F_{ij} as the confidence score of the pairwise constraint between x_i and x_j . To incorporate F into the subsequent clustering process, as in [5, 14], we adjust the learnt similarities between the data points according to the following similarity refinement formula

$$W_{ij}^* = \begin{cases} 1 - (1 - F_{ij})(1 - W_{ij}), & F_{ij} \geq 0; \\ (1 + F_{ij})W_{ij}, & F_{ij} < 0. \end{cases} \quad (12)$$

The above refinement can increase the similarity between x_i and x_j when $F_{ij} > 0$ and decrease it when $F_{ij} < 0$. Let W^* be the refined similarity matrix according to (12). We adopt the spectral clustering algorithm [18] with W^* to form final clusters.

4.2 Experimental setup

For comparison, the results of four notable and most related algorithms, Kamvar et al.'s spectral learning (SL) [7], Lu and Carreira-Perpinan's affinity propagation (AP) [12], Li and Liu's constrained clustering by spectral kernel learning [10] (CCSKL) and Lu and Ip's exhaustive and efficient constraint propagation (E²CP) [14], are also reported. In addition, we use the normalized cuts (NCuts) [17], which is effectively a spectral clustering algorithm but without considering pairwise constraints, as the baseline method.

In order to evaluate these algorithms, we compare the clustering results with the available ground-truth data labels, and employ the adjusted Rand (AR) index as the performance measure [6]. The AR index measures the pairwise agreement between the computed clustering and the ground-truth clustering, and takes a value in the range $[-1, 1]$. The larger is the adjusted Rand index, the better is a clustering result. To evaluate the algorithms under different settings of pairwise constraints, we exploit the ground-truth data labels and generate a varying number of pairwise constraints randomly for each dataset. That is, we randomly choose a pair of data points from each dataset. If they have the same class labels, we generate a must-link constraint, otherwise a cannot-link constraint. In the following experiments, we run these algorithms 20 times with random initializations, and report the averaged AR index.

Because these algorithms are all graph-based, we adopt the same k -NN similarity graph construction for all the algorithms to ensure a fair comparison. We empirically select $\lambda = 0.1$, $\mu = 0.1$, $\eta = 0.25$ and $k = 10$ in our experiments (we also analyze the influence of related parameters in our LSCP and other methods in Section 4.5). To get the original similarity measurements on each dataset, the spatial Markov kernel [13] is defined on the Scene datasets to exploit the spatial information, while the Gaussian kernel is used for the other three image datasets and the UCI datasets as in [10, 12]. All the algorithms are implemented in Matlab, running on a 2.33 GHz and 32GB RAM PC. For our LSCP, we use the standard QP solver *quadprog* in MATLAB to solve the QP problem.

4.3 Results on image datasets

In this experiment, we test the algorithms on four image datasets, USPS,¹ MNIST,² CMU PIE database (PIE), and a scene category dataset (Scene) [16]. USPS and MNIST contain images of handwritten digits from 0 to 9 of sizes 16×16 and 28×28 , respectively. There are 7291 training instances and 2007 test instances in USPS. MNIST has a training set of 60,000 instances and a test set of 10,000 instances. As in [10], for USPS, we use the total 9298 images from 0 to 9 digits, and for MNIST, we select a subset containing 5139 images from 0 to 4 digits in its test set. PIE contains 41,368 images of 68 people, each person under 13 different poses, 43 different illumination conditions, and with 4 different expressions. For PIE, we choose a subset from the first 10 individuals, which only includes five near frontal poses (C05, C07, C09, C27, C29) and all the images under different illuminations and expressions, thus there are 170 images for each individual and totally 1,700 images. We down-sample each image in PIE to 32×32 pixels. Scene dataset contains 8 scene categories, including four man-made scenes and four natural scenes (see Fig. 2). The total number of images in Scene is 2,688. The size of each image in this Scene dataset is 256×256

¹<http://www-stat.stanford.edu/tibs/ElemStatLearn/>

²<http://yann.lecun.com/exdb/mnist/>

Fig. 2 Sample images in the Scene dataset

pixels. For USPS, MNIST and PIE, the feature to represent each image is a vector formed by concatenating all the columns of the image intensities. For the Scene dataset, we extract the SIFT descriptors of 16×16 pixel blocks computed over a regular grid with spacing of 8 pixels. Then we perform k -means clustering on the extracted feature vectors to form a vocabulary of 600 visual keywords. Based on this visual vocabulary, we then define a spatial Markov kernel [13] for the construction of the similarity graph. We summarize the description of the above four image datasets in Table 1.

In the experiments, we compare the clustering performance of the six algorithms using different number of pairwise constraints. The clustering results on the four image datasets are shown in Fig. 3, from which we can find that our LSCP generally performs the best among the six related methods. After incorporating the pairwise constraints into clustering, the improvement achieved by our LSCP is very significant compared with NCuts, the baseline unconstrained method. The effectiveness of our approach, which firstly applies the initial pairwise constraints to learn a better similarity measurement and then propagates the initial pairwise constraints using the learnt graph, is verified by the fact that LSCP consistently obtains better results. As the number of constraints grows, the performance of our LSCP improves more significantly than that of the AP, SL and CCSKL. In contrast, AP and SL perform unsatisfactorily, and in some cases, their performances have been degraded to that of NCuts. As to CCSKL, while it also outperforms NCuts, but the most serious limitation of CCSKL is that its performance cannot benefit from the increasing number of the constraints. Especially, in some cases, more constraints even decrease the performance of CCSKL. Our LSCP is based on E²CP, and therefore like the E²CP, our LSCP achieves a unanimous and obvious improvement in the performance on all the four image datasets

Table 1 Description of the four image datasets

	USPS	MNIST	PIE	Scene
# objects	9298	5139	1700	2688
# dimensions	256	784	1024	600
# classes	10	5	10	8

Fig. 3 The clustering results on the four image datasets by different algorithms with a varying number of pairwise constraints

as the number of initial pairwise constraints increases. Moreover, due to the fact that our LSCP can apply the initial supervisory information, i.e. the pairwise constraints, to learn a new and better similarity measurement among the data points, our LSCP can get the better results than E²CP on each image dataset.

In addition, in Fig. 4, we compare our LSCP with the other two similarity learning methods, i.e. the linear neighborhood propagation (LNP) [20], which is a local similarity learning method, and constrained clustering by spectral kernel learning (CCSKL) [10], which is a global similarity learning method. Specifically, we firstly apply LNP and CCSKL to learn a similarity graph and then adopt E²CP as the basic constraint propagation method to propagate the pairwise constraints using the learnt similarity graphs by LNP and CCSKL, respectively. We respectively denote these constrained clustering results as LNP+PCP and CCSKL+PCP. From Fig. 4, we can see that our LSCP still consistently and significantly outperforms LNP+PCP and CCSKL+PCP on each image dataset. Like the CCSKL, CCSKL+PCP cannot improve its performance as the number of pairwise constraints increases. Moreover, in some cases, CCSKL+PCP even performs worse than CCSKL. As to LNP+PCP, although it can also benefit from the increasing of the number of constraints, but because the similarity graph learning process of LNP only considers the “local reconstruction error”, but does not takes any supervisory information, e.g. the pairwise constraints, into consideration, the performances of our LSCP are still significantly better than the LNP+PCP on each image dataset.

Fig. 4 The clustering results on the four image datasets by the three graph learning algorithms with a varying number of pairwise constraints

We also compare the three similarity graph learning methods, i.e. LNP, CCSKL and our LSCP, on the Scene image dataset by applying them to NCuts, SL and AP, which are all graph-based methods. We firstly learn a similarity graph and then respectively perform NCuts, SL and AP on the learnt similarity graph. The clustering results are shown in Fig. 5. From Fig. 5, we can find that our graph learning method LSCP consistently improve the performance of the three graph based algorithms, i.e. NCuts, SL and AP. The three graph learning methods all achieve better results than the original NCuts. And to SL, LNP and our LSCP not only outperform SL, but also improve the performance more significantly as the

Fig. 5 The clustering results on the Scene dataset obtained by applying three graph learning methods, i.e. LNP, CCSKL and our LSCP, to NCuts, SL and AP, respectively

number of constraints grows. When using the similarity graph learned by our LSCP, AP can also achieve better performance. The results in Fig. 5 illustrate the importance of learning a better similarity graph for improving the performance of the graph based methods.

To help visualize the effect of our LSCP which exploits the pairwise constraints for final constrained clustering, we show the distance matrices of the low-dimensional data representations obtained by NCuts, SL, AP, CCSKL, E^2 CP and LSCP in Fig. 6. We can find that the block structure of the distance matrices of the data representations obtained by LSCP on each image dataset is significantly more obvious, as compared to those of the data representations obtained by the other five methods. This means that after refining the similarities among the data by LSCP, each cluster associated with the new data representation becomes more compact and the different clusters become more separated.

It should be noted that the pairwise constraints used here are actually very sparse. For example, the total number of pairwise constraints that can potentially be defined for the USPS dataset with $n = 9298$ is 43,221,753, i.e. $(n^2 - n)/2$. In our experiments, for example, the used 2,400 pairwise constraints represent only about 0.0056 % of the total number of pairwise constraints that can be defined for the entire USPS dataset.

We also look at the time complexity of our LSCP on the different scale image datasets. For example, with 2,400 pairwise constraints, to learn the similarity graph on the Scene dataset (with 2,688 data points), our LSCP will take about 7.26 seconds; on the MNIST dataset (with 5,193 data points), our LSCP will take about 14.52 seconds; and on the USPS dataset (with 9,298 data points), our LSCP will take about 30.7 seconds. It can be viewed that the time complexity of our similarity learning process is $O(n)$ and our LSCP is a linear algorithm.

Fig. 6 Distance matrices of the low-dimensional representations for the four image datasets obtained by NCuts, SL, AP, CCSKL, E^2 CP and LSCP, respectively. The darker is a pixel, the smaller is the distance

Table 2 Description of the four UCI datasets

	Wine	Ionosphere	WDBC	Glass
# objects	178	351	569	214
# dimensions	13	34	30	10
# classes	3	2	2	7

4.4 Results on UCI datasets

We further conduct experiments on four UCI datasets, which are described in Table 2. The clustering results are shown in Fig. 7. Again, we can find that our LSCP performs the best. SL, AP and CCSKL that also consider the constraint information, have inconsistent performance on the four datasets. Especially, AP on Wine and Glass (or SL on Ionosphere) even does not perform better than NCuts. In addition, when the number of pairwise constraints grows, we can find a unanimous and obvious improvement in the performance of our LSCP on all the four datasets, but AP, SL and CCSKL do not present this trend in constrained clustering. In particular, on Ionosphere, more pairwise constraints even decrease the performance of SL and CCSKL. Similar to the situation on the image datasets, although

Fig. 7 The clustering results on the four UCI datasets by different algorithms with a varying number of pairwise constraints

Table 3 The performances (AR) of six methods with respect to different k values

	$k = 5$	$k = 10$	$k = 15$	$k = 20$	$k = 25$
NCuts	0.402	0.460	0.406	0.418	0.408
SL	0.426	0.441	0.430	0.426	0.403
AP	0.401	0.443	0.427	0.420	0.415
E ² CP	0.635	0.649	0.669	0.671	0.665
CCSKL	0.535	0.570	0.548	0.524	0.505
LSCP	0.680	0.685	0.676	0.677	0.675

the performances of E²CP also benefits from the increasing number of constraints, however because of the fact that it only makes use of the original and noisy similarity graph defined on the dataset, our LSCP that can apply the pairwise constraints to learn a better similarity graph consistently outperforms E²CP on each UCI dataset. To summarize, since the pairwise constraints can be exploited most effectively by our LSCP, its performance is the best.

4.5 The influence of parameters

We analyze the influence of parameters in our LSCP and other related methods in this section. In particular, we conduct experiments with different settings of k (the number of the nearest neighbors) for NCuts, SL, AP, E²CP, CCSKL and our LSCP on the Scene image dataset. We also conduct experiments with different settings of λ , μ and η for our LSCP on the Scene image dataset.

Firstly, we perform experiments with different k values and show the results of the six related methods in Table 3, where we apply 2,400 initial pairwise constraints and fix $\eta = 0.25$ for E²CP and our LSCP, and $\lambda = 0.1$ and $\mu = 0.1$ for our LSCP. It can be seen that among the five methods that consider the constraint information, our LSCP is the most stable method for different settings of k . NCuts, SL, AP, CCSKL and our LSCP all achieve the best results when $k = 10$, while E²CP achieves the best result when $k = 20$. However, when $k = 20$, our LSCP still outperforms E²CP.

Finally, we report the performance of our LSCP with respect to different values of three optimization parameters λ , μ and η on the Scene image dataset in Table 4, where we use 2,400 initial pairwise constraints and fix $k = 10$. In the first row, we fix $\mu = 0.1$ and $\eta = 0.25$, and change the value of λ from 0.1 to 1.0. In the second row, we fix $\lambda = 0.1$ and $\eta = 0.25$, and change the value of μ from 0.1 to 1.0. In the third row, we fix $\lambda = 0.1$ and $\mu = 0.1$, and change the value of η from 0.1 to 1.0. It can be seen that our LSCP consistently performs very stable under different settings of λ , μ and η .

Table 4 The performances (AR) of our LSCP with respect to different settings of the three parameters, i.e. λ , μ and η , in our algorithm

	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
λ	0.685	0.679	0.676	0.677	0.674	0.676	0.675	0.677	0.677	0.677
μ	0.685	0.685	0.688	0.688	0.688	0.689	0.690	0.690	0.691	0.691
η	0.676	0.677	0.687	0.688	0.690	0.690	0.689	0.689	0.686	0.686

5 Conclusions

In this paper, we have proposed a novel local similarity learning approach, which can exploit the supervisory information encapsulated in pairwise constraints, including must-link and cannot-link constraints, to learn a better similarity measurements between the instances. Our method is a locally learning algorithm, that is, we compute a new set of similarity measurements at each data point through simultaneously minimizing the local reconstruction error and the local constraint error. The learnt similarity graph is further used to propagate the initial sparse pairwise constraints across the entire dataset. Experimental results on a variety of real-life datasets have demonstrated the superiority of our proposed algorithm over the state-of-the-art techniques. Especially, from the experiments, it can be seen that the pairwise constraint propagation with the learnt similarity graph significantly outperforms the propagation with the original similarity graph, because the latter may suffer from the improper and noisy similarity measurements.

Acknowledgments We would like to thank the anonymous reviewers for their insightful suggestions. The work described in this paper is supported by National Science Foundation of China (No. 61272247, 61202231, 61300164, 61300165), Shanghai Science and Technology Committee (No. 13511500200), Beijing Natural Science Foundation of China (No. 4132037) and Ph.D. Programs Foundation of Ministry of Education of China (No. 20120001120130).

Open Access This article is distributed under the terms of the Creative Commons Attribution License which permits any use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

References

1. Basu S, Bilenko M, Mooney R (2004) A probabilistic framework for semi-supervised clustering. In: SIGKDD
2. Basu S, Davidson I, Wagstaff K (2009) Constrained clustering: advances in algorithms, theory, and applications. Chapman & Hall/CRC
3. Boyd S, Vandenberghe L (2004) Convex Optimization. Cambridge University Press
4. Chung F (1997) Spectral graph theory. Am Math Soc
5. Fu Z, Lu Z, Ip H, Peng Y, Lu H (2011) Symmetric graph regularized constraint propagation. In: AAAI
6. Hubert L, Arabie P (1985) Comparing partitions. J Classif 2(1):193–218
7. Kamvar S, Klein D, Manning C (2003) Spectral learning. In: IJCAI
8. Kulis B, Basu S, Dhillon I, Mooney R (2005) Semi-supervised graph clustering: a kernel approach. In: ICML
9. Lazebnik S, Schmid C, Ponce J (2006) Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In: 2006 IEEE computer society conference on computer vision and pattern recognition, vol 2, pp 2169–2178. IEEE
10. Li Z, Liu J (2009) Constrained clustering by spectral kernel learning. In: ICCV
11. Li Z, Liu J, Tang X (2008) Pairwise constraint propagation by semidefinite programming for semi-supervised classification. In: ICML
12. Lu Z, Carreira-Perpinán M (2008) Constrained spectral clustering through affinity propagation. In: CVPR
13. Lu Z, Ip H (2009) Image categorization by learning with context and consistency. In: CVPR
14. Lu Z, Ip H (2010) Constrained spectral clustering via exhaustive and efficient constraint propagation. In: ECCV
15. Lu Z, Ip H (2011) Spatial markov kernels for image categorization and annotation. IEEE Trans Syst Man Cybern Part B Cybern 41(4):976–989
16. Oliva A, Torralba A (2001) Modeling the shape of the scene: a holistic representation of the spatial envelope. Int J Comput Vis 42(3):145–175

17. Shi J, Malik J (2000) Normalized cuts and image segmentation. *IEEE Trans Pattern Anal Mach Intell* 22(8):888–905
18. von Luxburg U (2007) A tutorial on spectral clustering. *Stat Comput* 17(4):395–416
19. Wagstaff K, Cardie C (2000) Clustering with instance-level constraints. In: *ICML*
20. Wang F, Zhang C (2008) Label propagation through linear neighborhoods. *IEEE Trans Knowl Data Eng* 20(1):55–67
21. Wang X, Davidson I (2010) Active spectral clustering. In: *ICDM*
22. Wang X, Davidson I (2010) Flexible constrained spectral clustering. In: *SIGKDD*
23. Xing E, Ng A, Jordan M, Russell S (2003) Distance metric learning with application to clustering with side-information. In: *NIPS*
24. Zhou D, Bousquet O, Lal T, Weston J, Schölkopf B (2004) Learning with local and global consistency. In: *NIPS*
25. Zhu X, Goldberg A (2009) Introduction to semi-supervised learning. *Synth Lect Artif Intell Mach Learn* 3(1):1–130

Zhenyong Fu received the Ph.D. degree in the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China. He is currently an Assistant Professor in the College of Computer, Nanjing University of Posts and Telecommunications, Nanjing, China. His research interests includes machine learning, computer vision and multimedia processing.

Zhiwu Lu He received the M.S. degree in applied mathematics from Peking University in 2005, and the Ph.D. degree in computer science from City University of Hong Kong in 2011. He is currently an associate professor with the Department of Computer Science, School of Information, Renmin University of China. He has published over 40 papers in international journals and conference proceedings including *IJCV*, *TIP*, *TSMC-B*, *TMM*, *AAAI*, *IJCAI*, *ICCV*, *CVPR*, and *ECCV*. His research interests lie in machine learning, pattern recognition, and computer vision.

Horace H. S. Ip He received the B.Sc. (first-class honors) degree in applied physics and the Ph.D. degree in image processing from the University College London, London, U.K., in 1980 and 1983, respectively. Currently, he is the Chair Professor of computer science, the Founding Director of the Centre for Innovative Applications of Internet and Multimedia Technologies (AIMtech Centre), and the Acting Vice-President of City University of Hong Kong, Kowloon, Hong Kong. He has published over 200 papers in international journals and conference proceedings. His research interests include pattern recognition, multimedia content analysis and retrieval, virtual reality, and technologies for education. He is a Fellow of the Hong Kong Institution of Engineers, a Fellow of the U.K. Institution of Electrical Engineers, and a Fellow of the IAPR.

Hongtao Lu He received his Ph.D. degree in Electronic Engineering from Southeast University, Nanjing, in 1997. After graduation he became a postdoctoral fellow in Department of Computer Science, Fudan University, Shanghai, China, where he spent two years. In 1999, he joined the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, where he is now a professor. His research interest includes machine learning, computer vision and pattern recognition, and information hiding. He has published more than sixty papers in international journals such as IEEE Transactions, Neural Networks and in international conferences. His papers got more than 400 citations by other researchers.

Yunyun Wang She received the Ph.D. degree in Computer Science and Engineering from Nanjing University of Aeronautics and Astronautics, Nanjing, China. She is currently a lecturer in Nanjing University of Posts and Telecommunications. Her current research interests include pattern recognition, machine learning and neural computing.