Graph Based Constrained Semi-Supervised Learning Framework via Label Propagation over Adaptive Neighborhood

Zhao Zhang, *Member, IEEE*, Mingbo Zhao, *Member, IEEE*, and Tommy W.S. Chow, *Senior Member, IEEE*

Abstract—A new graph based constrained semi-supervised learning (G-CSSL) framework is proposed. Pairwise constraints (PC) are used to specify the types (intra- or inter-class) of points with labels. Since the number of labeled data is typically small in SSL setting, the core idea of this framework is to create and enrich the PC sets using the propagated soft labels from both labeled and unlabeled data by special label propagation (SLP), and hence obtaining more supervised information for delivering enhanced performance. We also propose a Two-stage Sparse Coding, termed TSC, for achieving adaptive neighborhood for SLP. The first stage aims at correcting the possible corruptions in data and training an informative dictionary, and the second stage focuses on sparse coding. To deliver enhanced inter-class separation and intra-class compactness, we also present a mixed soft-similarity measure to evaluate the similarity/dissimilarity of constrained pairs using the sparse codes and outputted probabilistic values by SLP. Simulations on the synthetic and real datasets demonstrated the validity of our algorithms for data representation and image recognition, compared with other related state-of-the-art graph based semi-supervised techniques.

Index Terms—Constrained semi-supervised learning, label propagation, adaptive neighborhood, sparse coding, soft-similarity measure, subspace learning

1 Introduction

In the real world, there are ever-increasing vision image data or non-vision text data generated in the Internet surfing and daily social communication. Most real data are high-dimensional and unlabeled (i.e., no labels to distinguish them) that are readily available. But the labeling process by researchers working on classifying or recognizing them is costly and time consuming. This is the major reason why semi-supervised learning (SSL) [1], [34] has been arousing considerable attention in machine learning and data mining research. The goal of SSL is to enhance the performance using supervised information of labeled data and their relationships to unlabeled data.

Based on clustering and manifold assumptions [1], [12] [17], that is, nearby points (or points on the same structure) are likely to have the same label, recent years have witnessed lots of efforts on the graph based SSL (G-SSL) [9], [10], [13], [14], [15], [16], [17], [18], [19], [28], [29], [31], [32], [33], [34], [35], [36], [37], [38], [39], [43] for its elegant mathematical formulation and effectiveness by mining the intrinsic (either local or global) geometrical structure inferred from both labeled and unlabeled data. G-SSL can be broadly divided into *transductive* and *inductive*. The inductive setting is mainly for classification based (either *local* [29], [31], [32], [35], [36], [37] or *global* [28], [39]) dimensionality reduction of high-dimensional data. But the localized methods usually involve the step of estimating optimal neighbor number k and kernel width, which is challenging in reality. The other setting is the label propagation (LP) that propagates the labels from labeled data to unlabeled data according to the distribution of both labeled and unlabeled data [9], [10], [13], [14], [15], [16], [17], [18], [19]. Three most popular LP methods are the harmonic function approach [16], the consistency method [17] and the recent special label propagation (SLP) [19]. Compared with the harmonic function and consistency methods, SLP can not only well detect outliers in data, but also output the labels as probabilistic values [18].

An important step of LP is to define an edge weight matrix W to measure the similarities between vertices in a faithful graph. A graph is usually constructed by finding the neighbors by k-neighborhood or ε-neighborhood [12]. One most popular weight assignment method for LP is Gaussian kernel similarity [17], but estimating an optimal kernel width is difficult [9], [15]. *Linear Neighborhood Propagation* (LNP) [9] was recently proposed to firstly approximate the whole graph by a series of overlapped linear neighborhood patches and the edge weights in each patch are then computed using the neighborhood linear projection. But LNP also suffers from the issue of setting fixed neighborhood size for each vertex and there is no reliable way to determine optimal k number. More recently, to achieve adaptive neighborhood for weight construction in LP, sparse coding (SC) based formulations have attracted many interests (e.g., [13], [14], [15], [18]). Given a data matrix $X = [x_1, x_2, \ldots, x_N] \in \mathbb{R}^{n \times N}$, a similar idea is to compute the (non-negative) sparse codes for each x_i individually from the l_1-norm minimization with X or the pre-calculated sparse neighbors of point x_i as the

- *Z. Zhang is with the School of Computer Science and Technology, Soochow University, Suzhou 215006, and the Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing 210023, China. E-mail: cszzhang@gmail.com.*
- *M. Zhao and T.W.S. Chow are with the Department of Electronic Engineering, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong. E-mail: {mzhao4, eetchow}@cityu.edu.hk.*

Manuscript received 18 Apr. 2013; revised 30 Oct. 2013; accepted 31 Oct. 2013. Date of publication 11 Dec. 2013; date of current version 3 Aug. 2015. Recommended for acceptance by F. Bonchi.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TKDE.2013.182

dictionary [13], [14], [15], [18]. These methods have two drawbacks. First, the sparse codes of each point are computed *individually* and there is no global constraint on the solution, so the global structures of data may be inaccurately captured in practice [2]. Second, if there are no sufficient clean data available, setting X itself as the dictionary may decrease the robustness to noise and outliers. To address this issue, we first propose a *Two-stage Sparse Coding* (TSC) approach to achieve adaptive neighborhood for weight assignment in SLP. The first stage recovers the possible corruptions or errors in data and constructs a clean informative dictionary by solving the principal component pursuit (PCP) problem [3], [4]. The second stage computes the sparse codes of all data vectors *jointly* by a l_1-norm minimization problem so that the global structures of data are captured.

In addition, considering that in SSL settings, the number of labeled data is typically small. So it is greater advantageous to use pairwise constraints (PC) than the class labels to reflect supervised information of data, since PC can be obtained by minimal effort and can provide more supervised information if there are enough points with labels available [28], [29], [30], [31]. But if the labeled number is too limited, the advantages of PC over the class labels will not exist any more. To address the insufficient data labeling issue, in this paper we construct the PC sets based on the propagated labels from both labeled and unlabeled data via SLP. Technically, we propose an adaptive neighborhood based SLP process induced pairwise constrained SSL framework, termed *Graph based Constrained Semi-Supervised Learning* (G-CSSL), for feature extraction and classification. Note that G-CSSL is more general, i.e., all existing LP and our TSC based SLP processes can be embedded into the proposed G-CSSL framework for predicting the labels of samples and constructing the PC sets. To obtain adaptive neighborhood, the sparse codes are used to define the edge weights in SLP process of our framework. In addition, to deliver enhanced inter-class separation and intra-class compactness, based on the outputted probabilistic values by SLP and sparse codes over both labeled and unlabled data, we also propose a voting strategy based *mixed soft-similarity measure* (MSM) for evaluating the similarity/dissimilarity of pairs in PC sets.

To the best of our knowledge, no prior study has been done in enriching the PC sets from the propagated soft labels via SLP for SSL, formulating a two-stage sparse coding for SLP, and proposing the MSM method from the propagated outputs and sparse codes over constrained pairs. Hence, this work has the potential to outperform the existing G-SSL methods and LP processes for data representation, image recognition and label prediction.

The paper is outlined as follows. Section 2 briefly reviews the SC problem and the existing weight assignment methods for LP. In Section 3, we propose the TSC approach to assign weights for SLP. Section 4 proposes the presented G-CSSL framework and the MSM method. Section 5 describes the settings and tests our techniques using synthetic and benchmark datasets. Finally, the paper is concluded in Section 6.

2 Preliminaries

We briefly review the SC problem and the existing weight assignments in LP, which are closely related to our work.

2.1 Sparse Coding Problem

Given a set of vectors $x_i,\ i = 1, 2, \ldots, N$, SC represents each x_i using as few points that most compactly expresses it from the data matrix $X = [x_1, x_2, \ldots, x_N] \in \mathbb{R}^{n\times N}$ as possible [20], [21]. Let $s_i = [s_{i,1}, \ldots, s_{i,i-1}, 0, s_{i,i+1}, \ldots, s_{i,N}]^{\mathrm{T}}$ be an n-dimensional coefficient vector, in which $s_{i,j}(i \neq j)$ is the contribution of each x_j for reconstructing x_i, then SC solves the sparse codes s_i of each x_i from the following problem:

$$s_i = \arg\min_{s_i} \|s_i\|_0, \text{Subj } x_i = Us_i, \tag{1}$$

where U with dictionary items represents a dictionary matrix and $\|\cdot\|_0$ is l_0-norm (i.e., the number of nonzero entries of a vector). Note that the above problem is difficult to solve due to the discrete nature of the l_0-norm. As a common practice, l_0 -norm is usually relaxed using l_1-norm. Hence the above problem can be converted into the following convex one in the presence of partial corruptions:

$$s_i = \arg\min_{s_i} \|s_i\|_1 + \lambda\|x_i - Us_i\|_1, \tag{2}$$

where λ is a positive parameter and $x_i - Us_i = e_i$ is an error term. The solution $S^* = \left[s_1^*, s_2^*, \ldots, s_N^*\right]$ to Eq. (2) is the "*sparsest representation*" of X [20], [21], [22], [23]. The standard SC computes the sparse codes of each point individually, i.e., there is no global constraint on its solution, so it may be inaccurate at capturing the global structures of data in reality and this drawback may depress its performance if available data is grossly corrupted [2]. Also, the matrix X itself is often set as the dictionary in many studies (e.g., [20], [21], [22], [23]), but if there are no sufficient clean data available, the dictionary is badly defined. As a result, the robustness of SC to noise and outliers will be greatly weakened [48]. So in noisy case, leaning a clean and informative dictionary is vital [5], [6], [7]. One most representative method is *K*-SVD [6] that is recently emerged as a powerful tool for dictionary learning and can work with any pursuit method, thereby tailoring the dictionary to the real-world applications.

2.2 Weight Assignment in Label Propagation

LP aims to propagate label information of labeled data to the unlabeled data according to the distribution of both labeled and unlabeled data [16], [17], [18], [19]. For LP, an undirected weighted graph is firstly constructed. Denote the graph as $\widehat{G} = (\widehat{V}, \widehat{E})$, where the vertex set $\widehat{V} = X$ and the edges in $\widehat{E}$ are weighted by $W_{i,j}$. There are various methods to define W for measuring the similarities between vertices, e.g., inverse euclidean distance (i.e., $W_{i,j} = \|x_i - x_j\|^{-1}$) [11] and Gaussian kernel (i.e., $W_{i,j} = \exp(-||x_i - x_j||^2/2\delta^2)$) [12], where $\|x_i - x_j\|$ is the euclidean distance between x_i and x_j. Another popular way is to use an idea similar to *Locally Linear Embedding* (LLE) [8], assuming that each data point can be reconstructed using a linear combination of its selected neighbors. But these methods suffer from the issue of determining optimal mode parameters, for instance kernel width and neighborhood size.

More recently, to avoid estimating optimal neighborhood size over each point, some works on SC based edge weight assignments for LP process were proposed [13], [14], [15]. In

this scenario, there are two major ways. The first one is to compute the sparse codes to assign weights [13], [15] by optimizing the following SC formulation:

$$s_i = \arg\min_{s_i} \|s_i\|_1, \text{Subj } x_i = Us_i, U = \widetilde{X_i}, s_i \geq 0, \quad (3)$$

where $\widetilde{X_i}$ is a data matrix obtained from X by removing x_i itself. The non-negative constraint can be omitted. For robust representation, the above problem requires sufficient noiseless data available in U. But setting $\widetilde{X_i}$ as the dictionary may put the validity and robustness of SC in jeopardy in reality and hence affecting subsequent weight construction in LP. Another representative method is *label propagation through sparse neighborhood* (LPSN) [14]. LPSN firstly solves Eq. (3) without the non-negative constraint to construct a sparse neighborhood $SN(x_i)$ of each x_i by involving a threshold ε that is not easy to estimate. LPSN then computes the weights $W_{i,j}$ by solving another SC problem by setting the sparse neighbors $SN(x_i)$ as dictionary when solving the sparse codes for each sample x_i. Finally, the labels of all points are estimated by

$$F = \arg\min_{F} \sum_{i=1}^{l+u} \left\| f_j - \sum\nolimits_{j:\, x_j \in SN(x_i)} W_{i,j} f_j \right\|^2, \quad (4)$$

where f_i is the predicted label of x_i, with the elements denoting the probabilities of x_i belonging to different classes. Note that the setting suffers from the same drawback as the above one. In addition, the probability values cannot be guaranteed to be nonnegative, which might violate the definition of probability.

3 Two-Stage Sparse Coding for SLP

This section proposes a two-stage sparse coding formulaton for achieving adaptive neighborhood for SLP to predict the labels of unlabeled samples. Then we can construct the PC sets based on the predicted soft labels. Let $X = [X_L, X_U] \in \mathbb{R}^{n\times(l+u)}$, where $X_L = [x_1, x_2, \ldots, x_l] \in \mathbb{R}^{n\times l}$ is a labeled set and $X_U = [x_{l+1}, x_{l+2}, \ldots, x_{l+u}] \in \mathbb{R}^{n\times u}$ is an unlabeled set. We assume that there are c classes and all classes are present in the labeled set. Each point in X_L is associated with a class label $l(x_i)$ in $\{1, 2, \ldots, c\}$. The special label propagation used in this work is *transductive*, propagating label information of X_L to X_U [19]. By assigning the sparse codes as weights, the predicted labels, termed *soft labels*, can be obtained by SLP.

3.1 Our Proposed Two-Stage Sparse Coding

Based on using matrix expressions, the SC problem can be transformed into the following non-convex problem with two variables, that is dictionary $U \in \mathbb{R}^{n\times N}$ and coefficients $S \in \mathbb{R}^{N\times N}$, involved together:

$$(U, S, E) = \arg\min_{U,S,E} \|S\|_1 + \lambda\|E\|_\ell, \text{Subj } X = US + E, diag(S) = 0, \quad (5)$$

subject to certain constraint on U, where $X - US$ identifies the errors E, and $diag(S) = 0$ is to avoid the trivial solution $S = I$ with I being an identity matrix. Note that $\|E\|_\ell$ is $l_{2,1}$ -norm, l_1-norm or squared Frobenius norm $\|\cdot\|_F^2$ of E, where $\|E\|_{2,1} = \sum_{j=1}^{N} \sqrt{\sum_{i=1}^{n} (E_{i,j})^2}$ is to model the sample-specific corruptions (and outliers), $\|E\|_1 = \sum_{i,j} |E_{i,j}|$ can be added for characterizing the random corruptions, and the squared Frobenius norm $\|\cdot\|_F^2$ is for modeling noise [2]. To solve this issue, a common approach is to alternately optimize U and S by minimizing over one variable while keeping the other one fixed [7], [24]. But due to the non-convex nature of the problem, the convergence property is difficult to be guaranteed. In this paper, we propose a *Two-stage Sparse Coding* to solve Eq. (5) by independently optimizing two convex subproblems. The first stage aims at correcting the corruptions in data and constructing a clean informative dictionary, while the second stage focuses on sparse coding with the constructed dictionary for representation and weight assignment.

3.1.1 Error Correction and Dictionary Construction

We first construct a clean informative dictionary U. Considering that most real observations usually include noisy data or missing values, or even are grossly corrupted [2], [3], [4], this work proposes to construct U (with the corruptions corrected and noise removed) from the following *Principal Component Pursuit* problem [3], [4]:

$$(U,\ E) = \arg\min_{U,E} \|U\|_* + \lambda\|E\|_\ell, \text{Subj } X = U + E, \quad (6)$$

where $X - U$ identifies the errors E, $\|\cdot\|_*$ is the nuclear norm of a matrix, i.e., the sum of the singular values of the matrix, and $l_{2,1}$-norm (or l_1-norm) can be imposed on E to model corruptions or outliers. This paper similarly sets $\lambda = \sqrt{\min(n, N)}/\sqrt{\max(n, N)}$ and chooses a sufficiently simple value in that range as [4]. Eq. (6) can be solved by using the inexact augmented lagrange multiplier (ALM) method [3]. By imposing l_1-norm on E, the augmented Lagrangian function of Eq. (6) is given as

$$J(U, E, Y_1, \mu) \doteq \|U\|_* + \lambda\|E\|_1 + \langle Y_1, X - U - E\rangle + \frac{\mu}{2}\|X - U - E\|_F^2, \quad (7)$$

where Y_1 is Lagrange multiplier, μ is a positive parameter and $\|\cdot\|_F$ is matrix Frobenius norm. Details can be referred to [3], [4], where U_k at each iteration k is solve by the singular value thresholding (SVT) operator [42] and E_k is solved by the shrinkage operator [3]. If $l_{2,1}$-norm is imposed on E, we can instead solve $Min_{U,E}\ \|U\|_* + \lambda\|E\|_{2,1}$ Subj $X = U + E$. After U_{k+1} is obtained at the *(k+1)*th iteration, the solution of E_{k+1} is inferred as

$$E_{k+1} = \arg\min_{E} (\lambda/\mu_k)\|E\|_{2,1} + \frac{1}{2}\left\|E - \left(X - U_{k+1} + Y_1^k/\mu_k\right)\right\|_F^2. \quad (8)$$

Note that this problem also has a closed-form solution [1]. After the optimal solution (U^*, E^*) is calculated, the original data can be recovered as U^* (or $X - E^*$). Note that the convergence of the inexact ALM based PCP has also been well studied [2], [3], [25]. Extensive image and video surveillance datasets have demonstrated the effectiveness of this recovery method in handling gross corruptions [3], [4], [44], [50]. Since the corruptions and noise in the original

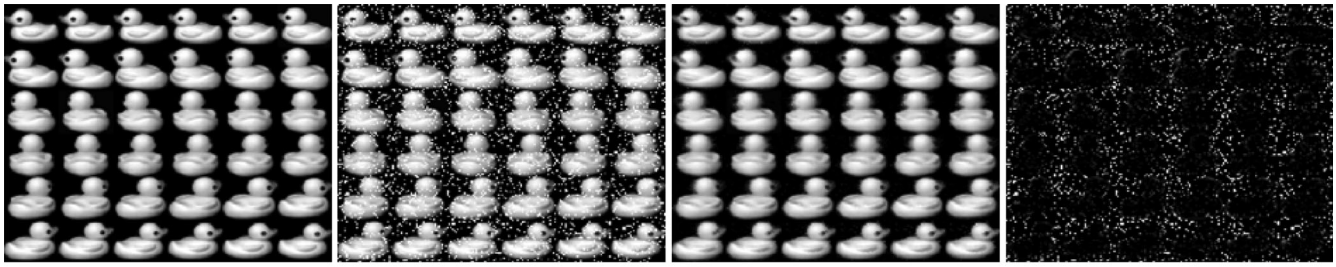


Fig. 1. Illustrating the advantages of U^*, where: (a) Original X, (b) Corrupted Φ, (c) Recovered U^*, and (d) Errors E^*.

data have been effectively recovered in U^*, the low-rank U^* will have the potential to be a dictionary. To show the advantages of using U^* as the dictionary, Fig. 1 gives an example, where we have shown the original data X, the corrupted data Φ and the recovered data U^*. Clearly, the corruptions can be accurately recovered by PCP. And intuitively, the low-rank principal component matrix U^* of the original data is more promising and appropriate than Φ being the dictionary in learning the sparse codes, which will be versified by simulations.

3.1.2 Sparse Coding and Weight Construction

With the dictionary U constructed, we are now ready to learn the sparse codes or reconstruction coefficients S for edge weight assignment. Since the possible errors and corruptions in data have been effectively recovered in the first step, we can solve the sparse codes S by optimizing the following convex problem:

$$S = \arg\min_{S} \|S\|_1, \text{ Subj } U = US,\ diag(S) = 0,\ S > 0,\ e^{\mathrm{T}}S = e^{\mathrm{T}}, \tag{9}$$

where $e \in \mathbb{R}^N$ is a column vector with all ones and the sum-to-one constraint $e^{\mathrm{T}}S = e^{\mathrm{T}}$ enables the sparsest presentation to preserve certain local information of data [20]. Here we learn the sparse codes of all vectors jointly, i.e., the global structures of data can be captured [2]. In this step, we also use the inexact ALM to solve Eq. (9). Firstly, it can be transformed to the following equivalent one:

$$\begin{aligned} &(Q, S) = \arg\min_{Q,S} \|Q\|_1 \\ &\text{Subj } S = Q,\ U = US,\ diag(S) = 0,\ S > 0,\ e^{\mathrm{T}}S = e^{\mathrm{T}}. \end{aligned} \tag{10}$$

The corresponding augmented Lagrangian function of the above problem can be similarly described as

$$\begin{aligned} &\widehat{J}(Q, S, Y_1, Y_2, \mu) \\ &\quad = \|Q\|_1 + \langle Y_1, U - US\rangle + \langle Y_2, S - Q\rangle + \frac{\mu}{2}(\|U - US\|_F^2 + \|S - Q\|_F^2). \end{aligned} \tag{11}$$

We first solve Q by fixing S. When solving Q_{k+1} at the *(k+1)*-iteration, both Y_2 and S are set to Y_2^k and S_k respectively. Thus, Q_{k+1} is inferred as $Q_{k+1} = \arg\min_Q (1/\mu_k)\|Q\|_1 + (1/2)\left\|Q - \left(S_k + Y_2^k/\mu_k\right)\right\|_F^2$, which can be solved by the shrinkage operator [3]. Then we can infer S_{k+1} as

$$S_{k+1} = \left(U^{\mathrm{T}}U + I\right)^{-1}\left[U^{\mathrm{T}}U + Q_{k+1} + \left(U^{\mathrm{T}}Y_1^k - Y_2^k\right)/\mu_k\right], \tag{12}$$

where U^{T} is the transpose of U and $\left(U^{\mathrm{T}}U + I\right)^{-1}$ is the inverse of $U^{\mathrm{T}}U + I$. Note that the solution S^* can be used to define the edge weight matrix W (i.e., $W = S^*$) for representing the sparsity of dataset and local information between vertices [27], that is, heavy weights $W_{i,j}$ will be imposed to the edges connecting "close" vertices. Ideally, intra-subject affinities will be denser than the inter-subject affinities that are all zeros [20], [21], [22]. Attributed to the nature of sparse representation, W has a natural discriminating power. In addition, to make a connection to the normalized graph, we first symmetrize W as $W \leftarrow (W + W^{\mathrm{T}})/2$ or $W_{i,j} \leftarrow \left(W_{i,j} + W_{j,i}\right)/2$. Then resembling [17], we normalize W as $\widehat{W} = D^{-1/2}WD^{-1/2}$ or $\widehat{W_{i,j}} = W_{i,j}/\sqrt{D_{ii}D_{jj}}$, where D with $D_{ii} = \sum_{j=1}^{l+u} W_{i,j}$ is a diagonal matrix. Note that this normalization can help strengthen the weights in low-density region and weaken the weights in high-density region, which is useful for handling the cases that the density of dataset varies dramatically [19].

3.2 SLP over Adaptive Neighborhood

Based on the normalized matrix $\widehat{W} = D^{-1/2}WD^{-1/2}$, we can predict the labels of unlabeled training data by SLP. Denote by $Y = [y_1, y_2, \ldots, y_{l+u}] \in \mathbb{R}^{(c+1)\times(l+u)}$ the initial labels of all samples. For the labeled data x_j, $y_{i,j} = 1$ if x_j belongs to the *i*-th class, otherwise $y_{i,j} = 0$; for unlabeled x_j, $y_{i,j} = 1$ if $i = c + 1$, otherwise $y_{i,j} = 0$. Note that SLP adds an additional class $c + 1$ to detect outliers, so the sum of each column of Y is 1 [19]. Also let $F = [f_1, f_2, \ldots, f_{l+u}] \in \mathbb{R}^{(c+1)\times(l+u)}$ be the predicted soft label matrix, where f_i is a column vector with entries $0 \le f_{i,j} \le 1$, and the biggest $f_{i,j}$ in each column decides the class assignment of sample x_i.

Denote a stochastic matrix $\Xi = \widehat{D}^{-1}\widehat{W}$, where $\widehat{D}$ is a diagonal matrix with each element being $\widehat{D_{ii}} = \sum_{j=1}^{l+u} \widehat{W_{i,j}}$. Then, we consider an iterative process for label propagation. At each iteration, SLP expects that the class label of each sample point is partially received from its neighborhoods and the rest is from its own label. Hence the label information of samples at the *(t+1)th* iteration can be

$$F(t + 1) = F(t)\Xi I_\alpha + YI_\beta, \tag{13}$$

where $I_\alpha \in \mathbb{R}^{(l+u)\times(l+u)}$ is a diagonal matrix with each input element being α_j, $I_\beta = I - I_\alpha$, α_j $\left(0 \le \alpha_j \le 1\right)$ is a parameter

for sample x_j to balance the initial label information of point x_j and the label information received from its neighbors during the iteration. According to [19], the regularization parameter α_j for the labeled sample x_j is set to α_l, and the parameter α_j for the unlabeled sample x_j is set to α_u in the simulations. Based on the above iterative process, we can have

$$F(t) = F(0)(\Xi I_\alpha)^t + YI_\beta \sum_{k=0}^{t-1} (\Xi I_\alpha)^k. \tag{14}$$

Based on the matrix properties, i.e., $\lim_{t\to\infty}(\Xi I_\alpha)^t = 0$ and $\lim_{t\to\infty} \sum_{k=0}^{t-1} (\Xi I_\alpha)^k = (I - \Xi I_\alpha)^{-1}$, therefore the iterative process of SLP can converge to

$$F = \lim_{t\to\infty} F(t) = YI_\beta (I - \Xi I_\alpha)^{-1}. \tag{15}$$

Note that it can be easily proved that the sum of each column in F is equal to 1, which indicates that the elements in F are the probability values and $f_{i,j}$ can be seen as the posterior probability of x_j belonging to the i-th class. If $i = c + 1$, $f_{i,j}$ represents the probability of x_j belonging to outliers. Based on the SLP process, the outliers in data can be detected and the soft labels of data can be obtained at the same time [19]. We describe the process of SLP over adaptive neighborhood in Algorithm 1.

Algorithm 1. Adaptive Neighborhood based SLP Process

Input: Data matrix $X \in \mathbb{R}^{n\times(l+u)}$.
Output: The predicted class label matrix $F \in \mathbb{R}^{(c+1)\times(l+u)}$.
1. Calculate the low-rank U for dictionary construction;
2. Learn the sparse codes via TSC to assign weights in W;
3. Symmetrize W as $W \leftarrow \big(W + W^{\mathrm{T}}\big)/2$ and normalize W as $\widehat{W} = D^{-1/2} W D^{-1/2}$, where D denotes a diagonal matrix with entries being $D_{ii} = \sum_{j=1}^{l+u} W_{i,j}$;
4. Construct a stochastic matrix as $\Xi = \widehat{D}^{-1}\widehat{W}$, where $\widehat{D}$ is a diagonal matrix being $\widehat{D_{ii}} = \sum_{j=1}^{l+u} \widehat{W_{i,j}}$;
5. Output the predicted soft label matrix $F = YI_\beta (I - \Xi I_\alpha)^{-1}$.

4 Graph Based Constrained Semi-Supervised Learning (G-CSSL)

We propose the G-CSSL framework mathematically in this section. The core idea is to create and enrich the PC sets using the propagated soft labels of data by the TSC based SLP process given in Section 3. We also address the mixed soft-similarity measure approach for similarity measurements. Next, we will begin with the introduction of the traditional constrained learning problem.

4.1 Traditional Constrained Learning Problem

In traditional pariwise constrained problem, for a given set of labeled data vectors, a *Must-link* (ML) constraint set and a *Cannot-link constraint* (CL) set [28], [29] are constructed as

$$ML = \big\{(x_i, x_j) | l(x_i) = l(x_j)\big\},\ CL = \big\{(x_i, x_j) | l(x_i) \neq l(x_j)\big\}, \tag{16}$$

where $l(x_i) \in \{1, 2, \ldots, c\}$ is the class label of x_i and c is the class number. Then one aims at pushing the data pairs $(x_i, x_j) \in ML$ close together in the reduced space by minimizing pairwise distances between them, and separating the data pairs $(x_i, x_j) \in CL$ via maximizing their pairwise distances. Thus we can define the following maximum margin criterion [38] based constrained problem:

$$\begin{aligned} \underset{T\in^{n\times d}}{Max} \frac{1}{2N_{CL}} \sum_{(x_i,x_j)\in CL} \big\|h(x_i) - h(x_j)\big\|^2 W_{i,j}^{(CL)} \\ - \frac{\vartheta}{2N_{ML}} \sum_{(x_i,x_j)\in ML} \big\|h(x_i) - h(x_j)\big\|^2 W_{i,j}^{(ML)}, \end{aligned} \tag{17}$$

where $h(x_i) = T^{\mathrm{T}} x_i$ is the low-dimensional representation of x_i, ϑ is a control parameter, N_{ML} and N_{ML} are the number of the ML and CL constraints respectively, $W^{(ML)}$ and $W^{(CL)}$ are the weight matrices for measuring pairwise similarities/dissimilarities over the ML and CL constraints. There are two popular ways (either *global* [28], [29] [39] or *local* [26], [30], [31]) to set the weights. In global setting, each sample pair $(x_i, x_j) \in ML$ or $(x_i, x_j) \in CL$ is equally treated (i.e., *hard-similarity measure*). In this case, $W_{i,j}^{(ML)} = 1$ for each $(x_i, x_j) \in ML$ and $W_{i,j}^{(CL)} = 1$ for each $(x_i, x_j) \in CL$; otherwise $W_{i,j}^{(ML)} = W_{i,j}^{(CL)} = 0$. The local setting incorporates local information of data into the definition of PC sets, and only neighbors from ML and CL sets are weighted with nonzero values; else zeros. In this scenario, either hard (e.g., *simple-minded method* [12]) or soft measure (e.g., *heat kernel* [12]) can be used. But these local settings also need to estimate optimal neighborhood size or kernel width.

It is also noted that the above problem is usually solved under a supervised setting, where the ML and CL sets are obtained from the ground-truth labels of data. Although PC can deliver some advantages over class labels, if the labeled number is too few, PC guided problems will have not superiority any more and even a disadvantage in special cases. For instance, if there are only two labeled data (either intra- or inter-class), we can only derive one single ML or CL constraint. To address the insufficient labeled data sampling problem, in what follows we will propose to solve the above problem under a semi-supervised setting, with the PC sets defined based on the propagated soft labels from both labeled and unlabeled data.

4.2 Our Proposed G-CSSL Framework

Based on the predicted soft label matrix F (where the entries of each column f_i are probabilistic values of each point belonging to different classes), the label of x_i can be assigned as $l(x_i) = \arg\max_{\Im \le c} F_{\Im,i}$. As a result, the insufficient data labeling issue can be naturally addressed. Then, based on the predicted soft labels, the enriched PC sets in this paper can be similarly constructed as

$$\begin{aligned} ML &= \big\{(x_i, x_j) | \arg\max_{\Im\le c} F_{\Im,i} = \arg\max_{\Im\le c} F_{\Im,j}\big\} \\ CL &= \big\{(x_i, x_j) | \arg\max_{\Im\le c} F_{\Im,i} \neq \arg\max_{\Im\le c} F_{\Im,j}\big\}. \end{aligned} \tag{18}$$

Note that the *ML* and *CL* constraint sets are defined over the first *c* classes from *F* in this framework, similarly as [40] that also used the first *c* rows of *F* for scatter matrix construction. This is mainly because the discovered novel class by SLP, i.e., the (*c+1*)th class, mainly include outliers or

ambiguous points from different classes that are difficult for classification. Since we have sufficient labeled points now, the superiority of PC over class labels can be highlighted to the greatest extent possible. In what follows, we first construct two weight matrices $W^{(ML)}$ and $W^{(CL)}$ of size $N \times N$ for similarity measurements via defining a voting strategy based mixed soft-similarity measure (MSM) before formulating the proposed G-CSSL framework.

4.2.1 Proposed Mixed Soft-Similarity Measure

The matrices $W^{(ML)}$ and $W^{(CL)}$ are firstly initialized with all zeros. Note that the sparse representation matrix S^* is naturally discriminant [20], [21], that is, it selects a set of samples that most compactly expresses given data and exclude other less compact points. So, if there are sufficient points from a subject, each point can be represented using a linear combination of points from a subject. Besides, the pairs, that contribute more together involving nonzero bigger $s^*_{i,j}$, are most likely to be "neighbors". Next we first symmetrize S^* as $S^* \leftarrow \left(S^* + S^{*\mathrm{T}}\right)/2$. Then for a given pair of instances $\left(x_i, x_j\right) \in ML$, this work assigns the following Cosine similarity based weights to measure their similarities:

$$W^{(ML)}_{i,j} = \begin{cases} \exp(s^*_{i,j}) \times \exp(\cos(\theta)), & if\ (x_i, x_j) \in ML \\ 0, & if\ (x_i, x_j) \notin ML \end{cases} \quad \text{with } \cos(\theta) = \frac{\langle f^\dagger_i, f^\dagger_j \rangle}{(\|f^\dagger_i\| \cdot \|f^\dagger_j\|)}, \tag{19}$$

where $\exp(\cdot)$ is exponential function, and $s^*_{i,j}$ is the *(i,j)*th entry of S^*. $f^\dagger_i$ denotes a column vector of the truncated version (i.e., $F^\dagger = [f^\dagger_1, f^\dagger_2, \ldots, f^\dagger_{l+u}] \in \mathbb{R}^{c\times(l+u)}$) of the predicted label matrix $F = [f_1, f_2, \ldots, f_{l+u}] \in \mathbb{R}^{(c+1)\times(l+u)}$, where $f^\dagger_i$ is the truncated f_i including the first *c* elements of f_i, and the entries of $f^\dagger_i$ satisfy $0 \le f^\dagger_{i,j} \le 1$. The Cosine similarity ($\in [0, 1]$) measures the similarity between two data vectors by computing the cosine of the angle between them. The main idea of the above weighting method is based on a voting strategy. Note that one major contribution of this paper is to create the pairwise constraints based on the propagated soft labels by SLP for delivering more supervised information. So ideally, if the predicted labels of samples by SLP are accurate, the data vectors $f^\dagger_i$ and $f^\dagger_j$ will be undoubtedly "close". As a result, the corresponding Cosine similarity (i.e., $\cos(\theta)$) is also higher. But note that under this condition we also consider information from the sparse codes to make the final decision for weight assignments, that is, a voting result is used. Two conditions are considered here. On one hand, if the predicted labels of data pair $\left(x_i, x_j\right) \in ML$ by SLP are accurate (i.e., $\cos(\theta)$ is bigger), and at the same time the samples x_i and x_j contribute more together (i.e., $\exp(s^*_{i,j})$ is bigger), heavier weight $W^{(ML)}_{i,j}$ will be incurred. On the other hand, we consider two cases where there exist ambiguous points. (1) We first consider the case where the ambiguous points from different classes are incorrectly predicted using LP to have the same label. In this case, $s^*_{i,j}$ may be zero or a very small number, i.e., $\exp(s^*_{i,j}) = 1$ or $\exp(s^*_{i,j}) \to 1$. So, a relatively lighter weight $W^{(ML)}_{i,j}$ will be incurred. (2) We consider another case where the ambiguous points of the same class are incorrectly predicted by LP to be different classes. In this case, $s^*_{i,j}$ may be large, but $\cos(\theta)$ is mall, thus a lighter weight $W^{(ML)}_{i,j}$ will also be incurred. In conclusion, the weight $W^{(ML)}_{i,j}$ is the heaviest if and only if $\exp(s^*_{i,j})$ and $\cos(\theta)$ are bigger at the same time, i.e., both the LP and SC processes reached a consensus. Hence the proposed weighting method is called voting strategy based mixed soft-similarity measure.

Based on similar idea, we define the following weights for each pair of instances $(x_i, x_j) \in CL$ predicted using SLP to measure the similarity between them:

$$W^{(CL)}_{i,j} = \begin{cases} \exp(1 - s^*_{i,j}) \times \exp(1 - \cos(\theta)), & if\ \left(x_i, x_j\right) \in CL \\ 0, & if\ \left(x_i, x_j\right) \notin CL. \end{cases} \tag{20}$$

Analogously, the heaviest penalty will be imposed on the edge weights $W^{(CL)}_{i,j}$ for $\left(x_i, x_j\right) \in CL$ if and only if both $\exp(1 - s^*_{i,j})$ and $1 - \cos(\theta)$ are bigger at the same time. Otherwise, a lighter penalty will be incurred.

4.2.2 The Objective Function of G-CSSL

After the weight matrices $W^{(ML)}$ and $W^{(CL)}$ are constructed, we can define the following objective function for our G-CSSL framework to calculate a projection matrix $T \in \mathbb{R}^{n\times d}$ onto which enhanced inter-class separation and intra-class compactness can be obtained at the same time:

$$\underset{T^{\mathrm{T}}T=I}{Max}\ \frac{1}{2}\sum_{i,j=1}^{N} \left\|h(\widehat{x_i}) - h(\widehat{x_j})\right\|^2 \widehat{W_{i,j}} + \frac{1}{2N_{CL}} \sum_{(x_i,x_j)\in CL} \left\|h(\widehat{x_i}) - h(\widehat{x_j})\right\|^2 W^{(CL)}_{i,j} - \frac{\vartheta}{2N_{ML}} \sum_{(x_i,x_j)\in ML} \left\|h(\widehat{x_i}) - h(\widehat{x_j})\right\|^2 W^{(ML)}_{i,j}, \tag{21}$$

where $\widehat{x_i} = U^* s^*_i$ denotes the reconstructed data with the learnt informative dictionary U^* and sparse coefficient vector s^*_i, $h(\widehat{x_i}) = T^{\mathrm{T}}\widehat{x_i}$ denotes the low-dimensional representation of data $\widehat{x_i}$, and $\widehat{W}_{i,j} = 1/N$ for each pair of instances. That is, $(1/2)\sum_{i,j=1}^{l+u} \left\|h(\widehat{x_i}) - h\left(\widehat{x_j}\right)\right\|^2 \widehat{W_{i,j}}$ is the *principal component analysis* (PCA) operator [45], thus this regularization $(1/2)\sum_{i,j=1}^{l+u} \|h(\widehat{x_i}) - h\left(\widehat{x_j}\right)\|^2 \widehat{W_{i,j}}$ is added to preserve the global covariance structures of all training samples including both labeled and unlabeled data, especially useful for the extreme case such that labels of all unlabeled data are incorrectly predicted by SLP to be *c*+1 class. In this extreme case, the number of constraints obtained only from labeled data is few, so the motivation for exploiting unlabeled data is to combine them for enhancing performance. Note that the above modeling is similar to the problem of [28], but we used the reconstructed data $\widehat{x_i}$ and our soft-similarity weights. We have a concise form for Eq. (21):

$$\underset{T^{\mathrm{T}}T=I}{Max}\ \frac{1}{2}\sum_{i,j=1}^{N} \left\|h(\widehat{x_i}) - h\left(\widehat{x_j}\right)\right\|^2 \Omega^{(CL)}_{i,j} - \frac{\vartheta}{2N_{ML}} \sum_{(x_i,x_j)\in ML} \left\|h(\widehat{x_i}) - h\left(\widehat{x_j}\right)\right\|^2 W^{(ML)}_{i,j}$$

$$where\ \Omega^{(CL)}_{i,j} = \begin{cases} \widehat{W_{i,j}} + (1/N_{CL}) W^{(CL)}_{i,j} & if\ \left(x_i, x_j\right) \in CL \\ \widehat{W_{i,j}} & if\ \left(x_i, x_j\right) \notin CL, \end{cases} \tag{22}$$

Since $\|T^{\mathrm{T}}\widehat{x}_i - T^{\mathrm{T}}\widehat{x}_j\|^2 = tr[(T^{\mathrm{T}}\widehat{x}_i - T^{\mathrm{T}}\widehat{x}_j)(T^{\mathrm{T}}\widehat{x}_i - T^{\mathrm{T}}\widehat{x}_j)^{\mathrm{T}}]$, based on the matrix form, the above problem becomes

$$\underset{T\in^{n\times d}}{Max}\ tr[T^{\mathrm{T}}\widehat{X}(L^{(CL)} - \vartheta L^{(ML)})\widehat{X}^{\mathrm{T}}T],\ \text{Subj}\ T^{\mathrm{T}}T = I, \quad (23)$$

where $L^{(ML)} = D^{(ML)} - W^{(ML)}$ and $L^{(CL)} = \Theta^{(CL)} - \Omega^{(CL)}$ are graph Laplacian matrices, $W^{(ML)}_{i,j} = (1/N_{ML})W^{(ML)}_{i,j}$, $D^{(ML)}$ and $\Theta^{(CL)}$ are diagonal matrices with the entries being $D^{(ML)}_{ii} = \sum_j W^{(ML)}_{i,j}$ and $\Theta^{(CL)}_{ii} = \sum_j \Omega^{(CL)}_{i,j}$, $tr(\cdot)$ denotes the trace operator and $\widehat{X} = [\widehat{x_1}, \widehat{x_2}, \ldots, \widehat{x_N}]$. ϑ is a control parameter for trading-off $tr(\widehat{X}L^{(CL)}\widehat{X}^{\mathrm{T}})$ and $tr(\widehat{X}L^{(ML)}\widehat{X}^{\mathrm{T}})$, where $tr(\widehat{X}L^{(CL)}\widehat{X}^{\mathrm{T}})$ can measure the separation degree of points and $tr(\widehat{X}L^{(ML)}\widehat{X}^{\mathrm{T}})$ measures the compactness degree of samples. From Eq. (23), the projection matrix $T \in^{n\times d}$ can be obtained including the orthogonal eigenvectors according to leading d eigenvalues of the eigen-problem: $\widehat{X}(L^{(CL)} - \vartheta L^{(ML)})\widehat{X}^{\mathrm{T}}\psi_j = \widehat{\lambda}_j\psi_j$. After T is obtained, dimensionality reduction of the dataset X can be performed in the form of $T^{\mathrm{T}}X$ and the projection matrix can be used for embedding new data in classification. More specifically, when a new test data is input, its low-dimensional embedding can be obtained by projecting it onto the projection axes in T. Note that an early version appeared in [47]. This extension also presented a two-stage SC to gain adaptive neighborhood for SLP and conducts a thorough evaluation on classification. We summarize the G-CSSL framework in Algorithm 2.

Algorithm 2. The proposed G-CSSL framework

Input: Data matrix $X \in \mathbb{R}^{n\times N}$ including labeled set X_L and unlabeled set X_U;
The reduced dimensionality $d \leq n$.
Output: The transformation matrix $T \in \mathbb{R}^{n\times d}$.
1. Predict the soft labels of samples using Algorithm 1;
2. Construct PC sets from the propagated labels and define the MSM using propagated outputs and sparse codes;
3. Solve the eigen-problem: $\widehat{X}\big(L^{(CL)} - \vartheta L^{(ML)}\big)\widehat{X}^{\mathrm{T}}\psi_j = \widehat{\lambda}_j\psi_j$, where $T \leftarrow [\psi_1, \psi_2, \ldots., \psi_d]$ according to d leading eigenvalues.

4.3 Comparison of Our Work and Existing Studies

We discuss the important issues related to the major contributions of this paper. Compared with the other related studies, this work can exhibit the following properties:

1) *Pairwise constrained SSL framework.* Virtually all previous pairwise constrained SSL methods (e.g., [28], [29] [31], [39]) are guided by the constraints constructed from the labels of labeled data. But labeled data is always expensive to obtain in reality, so available supervised information is often limited. In contrast, our G-CSSL is formulated based on enriching the PC sets from the propagated labels of both labeled and unlabeled data via SLP, so the above shortcoming can be effectively addressed.
2) *SC based weight assignment for LP.* In previous studies, the sparse codes are usually solved from Eq. (3) by setting X as the dictionary. But most real data is noisy or (grossly) corrupted, so this operation may directly affect the validity and robustness of SC in reality. But on the contrary, our proposed two-stage sparse coding for assigning weights in SLP can overcome the above issue effectively, because our TSC involves a separate step to correct the possible corruptions in data and learn a clean dictionary before computing the sparse codes.
3) *Similarity measurement between data points in PC guided problems.* In Section 4.1, we have shown the existing (either *soft* or *hard*) measurement methods. But both soft and hard measures have certain shortcomings, e.g., hard methods fail to consider the similarity difference among different pairs and soft measures usually involve the operation of determining neighborhood size or kernel width that is never easy in reality. In contrast, our proposed MSM uses the probabilistic soft labels by SLP and sparse codes to set the weights in a soft manner, which does not need to estimate the above model parameters. More importantly, MSM is based on a voting strategy, i.e., the final decision for assigning weights are determined by a voting result of the SLP and SC processes.

5 Simulation Results and Analysis

This section tests our G-CSSL algorithm, along with illustrating the results. In this study, we mainly examine our G-CSSL for image feature extraction and representation. Since G-CSSL is a SLP process induced PC based SSL algorithm, its performance is mainly compared with *semi-supervised dimensionality reduction* (SSDR) [28], *Semi-Supervised Metric Learning* (SSML) [29], *Marginal Semi-Supervised Sub-Manifold Projections* (MS^{3}MP) [26], *orthogonal MS^{3}MP* (OMS^{3}MP) [26] and *Semi-supervised Orthogonal Discriminant Analysis* (SODA) [40]. Note that SSDR, SSML, MS^{3}MP and OMS^{3}MP are pairwise constrained SSL algorithms, while SODA performs SSL via label propagation. SSML, MS^{3}MP, OMS^{3}MP and SODA have a common parameter (i.e., neighborhood size *k*) to estimate. In addition, SLP uses the Gaussian kernel to assign edge weights, so it has a parameter (i.e., kernel width δ) to estimate. To provide a reasonable estimation for δ, the kernel width is defined as $\delta = \widehat{\delta}/\varpi$, $\widehat{\delta} = \sum_{i,j}\|x_i - x_j\|^2/(N^2 - N)$ with a carefully chosen ϖ, similarly as [19], [41]. For *k*-neighbor search based methods, the *k* number is carefully tuned from $\{5, 7, 9, 11\}$ and the best performance is reported. The l^1-norm is imposed on the error term *E* in our TSC setting. We perform all simulations on a PC with Intel (R) Core (TM)2 Quad CPU Q9550 @ 2.83 GHz 2.83 GHz.

For TSC based SLP (TSC-SLP), label prediction process is the same as SLP except for the weight assignment. The recognition process of our G-CSSL and other methods are described as follows. Each dataset is randomly split into a training set X_{Tr} and a test set X_{Te}. The training set including labeled data X_L and unlabeled data X_U is used to train a learner. Prior to subspace learning, PCA is used to eliminate the null space of training set. The data X_{Te} is then embedded onto the reduced output space with the projection matrix learned from training data. Finally, the learner is used to evaluate the accuracies of test set. The

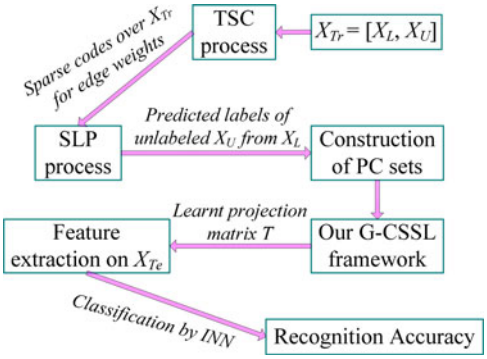


Fig. 2. Pattern recognition procedures of TG-CSSL.

one-nearest-neighbor classifier with euclidean metric is used for classification due to its simplicity. We show the procedures of applying our TSC-SLP based G-CSSL (TG-CSSL) algorithm for pattern recognition in Fig. 2.

In this study, one synthetic dataset and two real problems are tested. The first one is a "two moon" dataset; the second one is COIL-20 database (available at *http://www.cs.columbia.edu/CAVE/software/softlib/coil20.php*); the third one is COIL-100 database (available at *http://www.cs.columbia.edu/CAVE/software/softlib/coil100.php*). As is common practice, all images are resized to 20×20 pixels due to the computational consideration, thus each image corresponds to a point in a 400-dimensional space.

5.1 Toy Problem

We first compare our TSC-SLP process with the Gaussian kernel based SLP process using a toy problem in terms of robustness to the model parameters. We generate a toy dataset that includes two classes, each of which follows a half-moon manifold or distribution. In each class, two data points are selected as labeled set and the remaining as unlabeled set. The labels of unlabeled set are then predicted by SLP and our TSC-SLP. Fig. 3 illustrates the tansduction results of SLP on this toy. We see clearly that the results of SLP are sensitive to the kernel width δ of the Gaussian function for this toy. We in Fig. 4 show the tansduction results of our TSC-SLP with different selections of α_u, where α_l is fixed to 0. As can be observed, our TSC-SLP can perform well in a wide range of α_u. In other words, we experimentally find that TSC-SLP is less sensitive to the model parameter α_u. Apparently, our TSC-SLP setting that is insensitive to the model parameter will be more applicable for real-world problems.

We also compare our TG-CSSL with the standard SLP based SODA algorithm for classification on the toy problem. Because the distribution of the "two moon" dataset is non-Gaussian, we use the KPCA-trick [46] for kernelizing TG-CSSL and SODA, i.e., KPCA is firstly used to preprocess the dataset and them perform TG-CSSL and SODA for dimensionality reduction. Fig. 5 illustrates the gray images of the reduced output space learned by our TG-CSSL and SODA, in which the value of each pixel in images represents the distance difference from a pixel to its nearest labeled points after dimension reduction using TG-CSSL and SODA. In this simulation, the dimension of the learn projection is fixed to 1. From Fig. 5, we can observe that our TG-CSSL algorithm is cable of achieving a more desired classification decision boundary than SODA in accurately capturing the true distribution of the dataset, which is mainly because

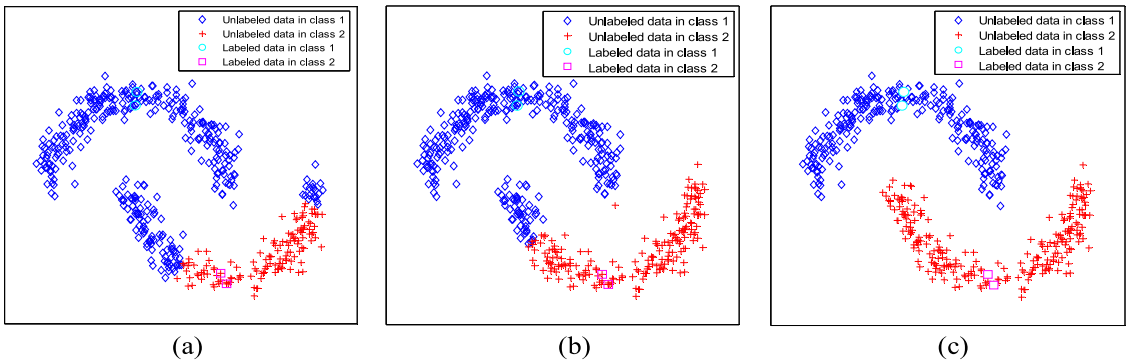


Fig. 3. Transduction results on the "two-moon" dataset using the Gaussian kernel based SLP. This figure illustrates the performance variation with various widths δ. (a) $\delta = \widehat{\delta}/0.1$, (b) $\delta = \widehat{\delta}$, and (c) $\delta = \widehat{\delta}/10$.

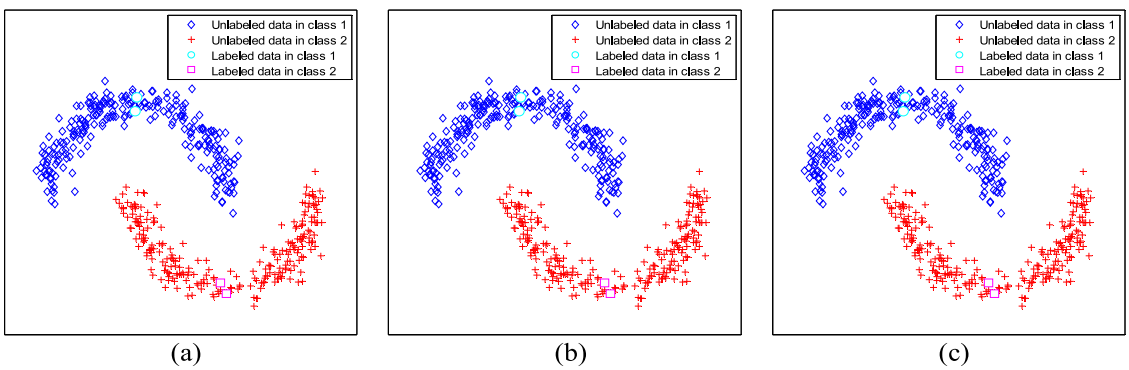


Fig. 4. Transduction results on the "two-moon" dataset using our TSC-SLP. This figure illustrates the performance variation with various parameters α_u. (a) $\alpha_u = 0.999$, (b) $\alpha_u = 0.999999$, and (c) $\alpha_u = 0.999999999$.

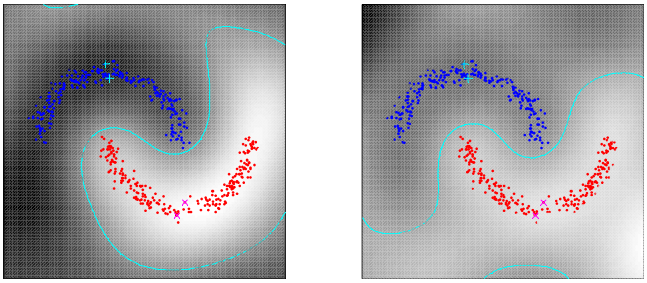


Fig. 5. The gray images of the reduced space by SODA (left) and our TG-CSSL (right) on "two moon" dataset.

TG-CSSL determines the nergborhood of each data point adaptively and defines the mixed soft-similarity seasure which contributes to enhancing the inter-class separation and intra-class compactness.

5.2 Object Recognition on COIL-20 Database

We evaluate our TG-CSSL algorithm to recognize the objects from COIL-20 database. This database has a total of 1,440 gray object images with black background for 20 different subjects, with 72 images per subject. We mainly compare our G-CSSL framework with SSDR, SSML, MS^{3}MP, OMS^{3}MP and SODA. In the experiments, the PC sets in SSDR and SSML are created based on whether labels of samples in X_L are the same or different [28], [29], while the PC sets are obtained relying on whether the labels of neighboring points in X_L are the same or not in MS^{3}MP and OMS^{3}MP [26]. For fair comparison, the labels of unlabeled data are predicted by SLP for SODA and our method. To show the superiority of our TSC approach over SC for edge weight construction, we also compare our TG-CSSL technique with the SC process based G-CSSL (G-CSSL) algorithm, where the sparse codes computed from the following SC problem are assigned to the edge weight matrix W:

$$\begin{aligned}(S,E) &= \arg\min_{S,E} \|S\|_1 + \lambda\|E\|_1 \\ \text{Subj } X &= XS + E, diag(S) = 0, S > 0, e^T S = e^T,\end{aligned} \tag{24}$$

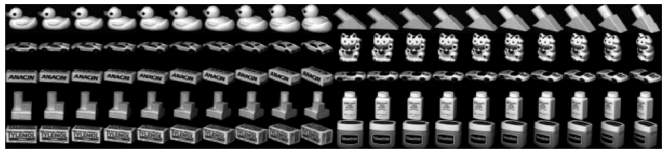


Fig. 6. Typical samples of the first 10 objects in COIL-20.

where X itself is set to be the dictionary and the solution can be similarly obtained as Eq. (9). In the simulations, the parameters ϑ in our G-CSSL and λ in SC are carefully selected from $\{10^i | i = -6, -5, \ldots, 6\}$ for fair comparison and the best results are reported. In addition, we also add the K-SVD approach [6], that is proposed to design overcomplete dictionary for SC, to be compared with. For fair comparison, K-SVD uses the same framework (i.e., our G-CSSL) for subspace learning. Note that SC based G-CSSL and K-SVD based G-CSSL perform classification in a similar way to the procedures in Fig. 1, except for the sparse coding and weight assignment processes. In the simulations below, the regularization factor α_l is set to 0 and α_u is chosen from $\{1 - 10^{-i} | \ i = 3, 5, \ldots, 15\}$ for SLP. To keep consistent with the size of the dictionary in SC and our TSC, we always used the K-SVD method learning a dictionary of size $n \times N$. The MATLAB code of K-SVD from *http://www.cs.technion.ac.il/~elad/software/* is used and we execute K-SVD over 20 iterations.

5.2.1 Visualization of the Graph Adjacency Matrices

This study visually compares the adjacency matrix of the TSC based l^1 -graph with the k-neighbor graph and the SC based l^1 -graph. The first 10 objects of the COIL-20 database are selected and the first 40 images per object are chosen for this study. Fig. 6 shows typical images of the 10 objects. For the k-neighbor graph, the k value is set to 7. The delivered adjacency matrices of the three graphs are illustrated in the first row of Fig. 7.

For easy comparison, we also exhibit the zooming in of the rectangle boxes in the second row of Fig. 7. We can observe from the results that the locality of the images is

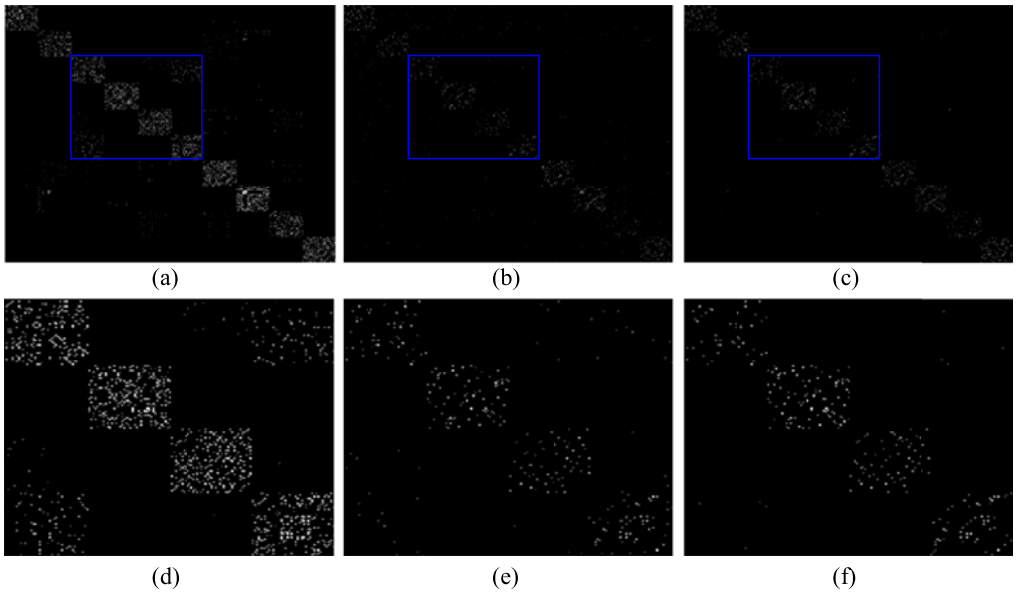


Fig. 7. Visualization of the adjacency matrix of (a) k-neighbor graph, (b) SC based l^1-graph, and (c) TSC based l^1-graph.

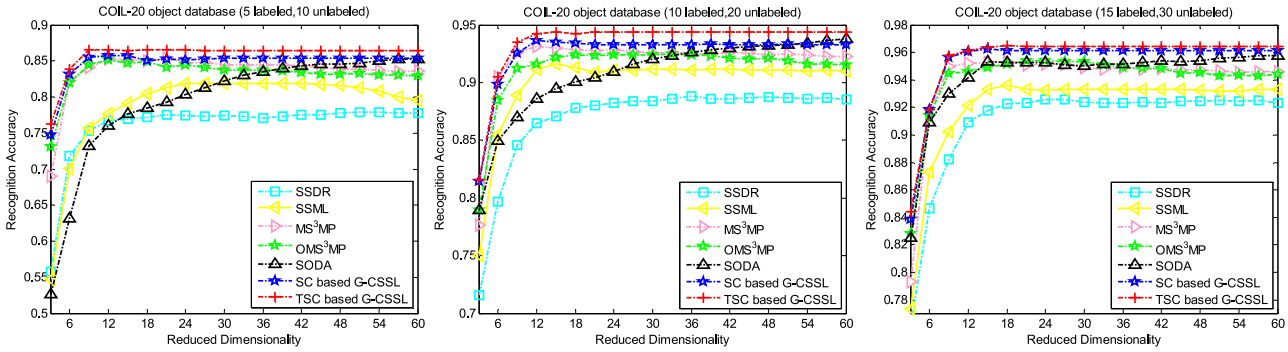


Fig. 8. Recognition accuracy versus different reduced dimensions on the COIL-20 database.

better encoded using the SC based l^1-graph and our TSC based l^1-graph, and more discriminant information has been delivered. That is, the l^1-graph can implicitly emphasize the natural clusters with each object and are able to succeed in identifying more "locality" of the images within the same object class. On the contrary, more inter-object connections are produced by the k-neighbor graph, but this observation may directly affect the subsequent LP process for predicting the labels of data and may also result in congregating the embeddings of inter-object images in the reduced space, and hence decreasing the performance. By comparing with the SC based l^1-graph adjacency matrix, the number of inter-object connections is further reduced by our TSC which learns the sparse codes with a clean and informative dictionary.

5.2.2 Object Recognition

This simulation evaluates our TG-CSSL by object recognition. The performance is mainly compared with SSDR, SSML, MS^{3}MP, OMS^{3}MP, SODA and G-CSSL. Three experimental settings over various numbers of labeled data (i.e., 5, 10 and 15 labeled respectively) randomly selected from each object class are tested. For each case, the number of unlabeled data is double the number of labeled data. For each setting, we regulate the numbers of reduced dimensions from 3 to 60 with interval 3, and the results are averaged over first 15 best records based on 20 realizations of training/test sets. We illustrate the results in Fig. 8, in which the simulation settings are also described. For fair comparison, SSDR, SSML, MS^{3}MP and OMS^{3}MP also use all available constraints to learn the projections. We find that: (1) In each setting, the performance of each algorithm goes up with the increasing numbers of reduced dimensions. (2) When the numbers of labeled and unlabeled data increase, the performance of SODA increases faster than SSDR and SSML in most cases. MS^{3}MP and OMS^{3}MP can outperform SODA, SSDR and SSML for recognizing the objects. SSDR is always the worst in each setting on this dataset. (3) Based on the reasonable formulations, our TG-CSSL can deliver higher accuracy than other techniques across all d values in most cases. Although SODA also uses LP to predict labels of unlabeled data for obtaining more supervised information to boost the performance, the major reason for the superiority of G-CSSL and TG-CSSL over SODA is due to the advantages of the pairwise constraints over class labels to some extent, that is, more supervised information can be obtained using the pairwise constraints as enough samples with labels are available.

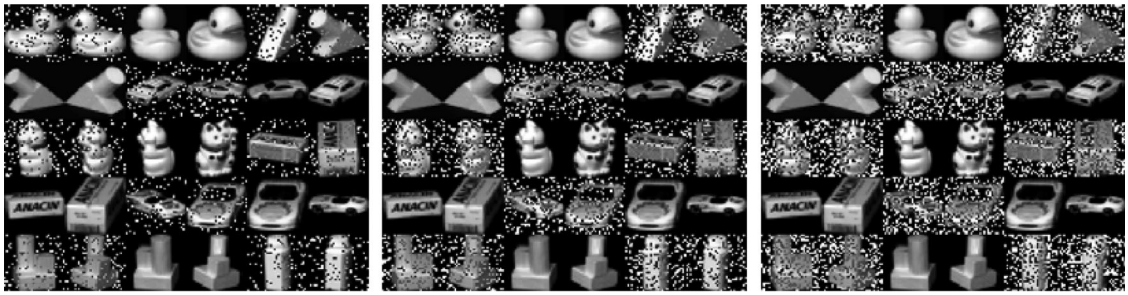
In Table 1, we summarize the statistics according to Fig. 8, including the mean accuracy, best record and the optimal image subspace (i.e., *Dim*), where the optimal subspace corresponds to the highest recognition accuracy of each algorithm in each setting. The results of KSVD based G-CSSL are also described. We have the following similar observations. First, the accuracies of all algorithms are improved as the number of training data increases. Second, our TG-CSSL can always achieve comparable and even better accuracies than other methods. And most importantly, our TG-CSSL is able to deliver the best results using smaller number of reduced dimensions in each case. Note that G-CSSL and KSVD based G-CSSL also perform better in most cases. The major reason may owe to the adaptive neighborhood and noise removal by SC.

5.2.3 Object Recognition against Pixel Corruptions

We also address an experiment to examine the robustness of our TG-CSSL algorithm in recognizing the objects

TABLE 1
Performance Comparison of the Algorithms on the COIL-20 Object Database

Result/Method	COIL-20 (5 labeled)			COIL-20 (10 labeled)			COIL-20 (15 labeled)		
	Mean	Best	Dim	Mean	Best	Dim	Mean	Best	Dim
SSDR	0.7597	0.7886	51	0.8683	0.8881	36	0.9090	0.9259	24
SSML	0.7886	0.8193	36	0.8993	0.9161	15	0.9201	0.9364	18
MS^{3}MP	0.8352	0.8538	21	0.9176	0.9312	12	0.9397	0.9537	18
OMS^{3}MP	0.8320	0.8515	12	0.9121	0.9248	33	0.9405	0.9540	24
SODA	0.7943	0.8529	60	0.9077	0.9378	60	0.9432	0.9578	57
SC based G-CSSL	0.8476	0.8578	12	0.9253	0.9361	12	0.9528	0.9627	15
KSVD based G-CSSL	0.8526	0.8623	18	0.9306	0.9426	21	0.9530	0.9651	15
TSC based G-CSSL	0.8582	0.8657	12	0.9347	0.9438	15	0.9554	0.9648	18



(a) 10% pixels corrupted (b) 20% pixels corrupted (c) 40% pixels corrupted

Fig. 9. Typical samples of the original images and corrupted images under various levels.

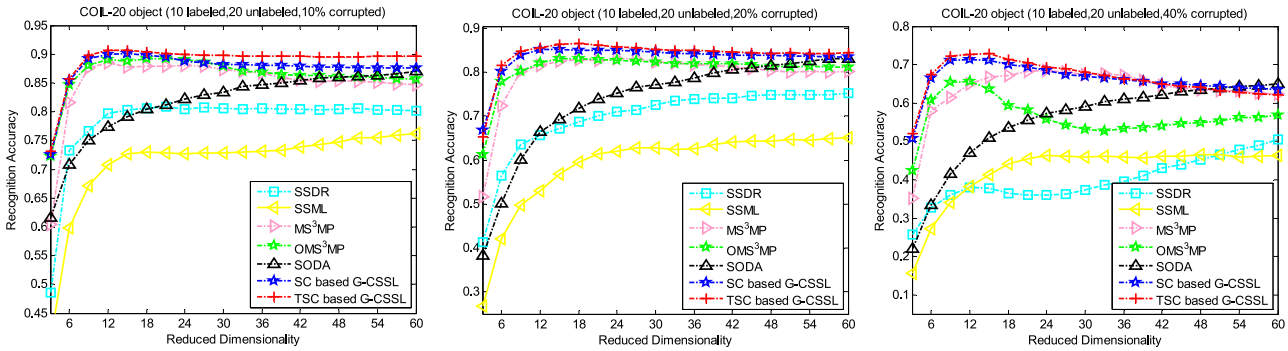


Fig. 10. Recognition accuracy versus different reduced dimensions on COIL-20 database with pixel corruptions.

under various degrees of random pixel corruptions. This study considers three settings over different levels of corruptions: one is with 10 percent pixels corrupted, one is with 20 percent pixels corrupted and the last one is with 40 percent pixels corrupted. For each pixel selected to be corrupted, its pixel value ξ is replaced by its inverse pixel, i.e., subtracting ξ from the biggest pixel value of images. We show typical samples in the training set, including the original images and corrupted images under various levels in Fig. 9. The number of labeled data per object is fixed to 10 and the number of unlabeled data is still double the labeled number. To investigate the robustness of each method against pixel corruptions, half of the labeled set, half of the unlabeled set and the whole test set are corrupted. In each setting, the first half of the labeled data (and unlabeled data) from each object class are chosen to be corrupted. For each method, the training set, including both labeled and unlabeled data, is applied to train the classifier and the test set is used for evaluation.

We show the averaged results of cases handing pixel corruptions in Fig. 10. We have the following findings. First, increasing the number of reduced dimensions can initially enhance the performance of each approach, but decrease the results of some methods after certain special point in each case. This is because at that special point, the dimensionality exceeded the "intrinsic" dimension of the training set. Second, the increasing level of pixel corruptions can decrease the recognition power of each approach. Specifically, SSDR and SSML are more sensitive to the pixel corruptions in images, since their accuracies decreased faster than other methods. SODA, MS^{3}MP, G-CSSL, and TG-CSSL are more robust against corruptions in all cases for their reasonable motivations and formulations. OMS^{3}MP works well in the first two cases, but as the level of pixel corruptions is increased to 40 percent, the performance of OMS^{3}MP is significantly weakened.

According to Fig. 10, we report the mean accuracy, highest accuracy and the optimal object image subspace in Table 2. We also describe the results of KSVD based

TABLE 2
Performance Comparison of the Algorithms on the COIL-20 Dataset with Pixel Corruptions

Result/Method	COIL-20 (10% corrupted)			COIL-20 (20% corrupted)			COIL-20 (40% corrupted)		
	Mean	Best	Dim	Mean	Best	Dim	Mean	Best	Dim
SSDR	0.7829	0.8085	21	0.6961	0.7512	60	0.3991 0.5054	0.5054	60
SSML	0.7104	0.7625	60	0.5890	0.6509	60	0.4229	0.4664	51
MS^{3}MP	0.8505	0.8857	12	0.7949	0.8303	24	0.6347	0.6852	24
OMS^{3}MP	0.8649	0.8930	21	0.8066	0.8305	18	0.5633	0.6560	12
SODA	0.8159	0.8705	60	0.7339	0.8299	60	0.5539	0.6505	60
SC based G-CSSL	0.8753	0.9018	15	0.8315	0.8510	15	0.6606	0.7129	12
KSVD based G-CSSL	0.8798	0.9035	18	0.8327	0.8525	18	0.6651	0.7291	15
TSC based G-CSSL	0.8883	0.9076	15	0.8385	0.8643	18	0.6641	0.7287	15

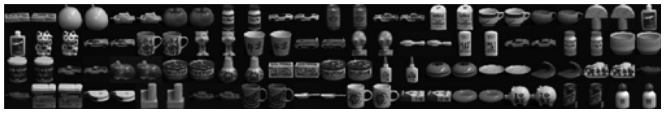


Fig. 11. Typical samples of first 50 objects in COIL-100.

G-CSSL. We find that: (1) The performance of each method goes down as the level of corruptions is increased. (2) Our TG-CSSL can outperform other methods in delivering the boosted accuracies in most cases. Our TG-CSSL method achieves comparable or even better results than KSVD based G-CSSL, and both are superior to G-CSSL. SODA obtains comparative highest records with G-CSSL, KSVD based G-CSSL and our TG-CSSL technique in most cases. Note that SSDR and SSML deliver the worst results in all cases. (3) G-CSSL, KSVD based G-CSSL and our TG-CSSL can achieve the highest records with smaller number of reduced dimensions involved in most cases.

5.3 Object Recognition on COIL-100 Database

This study examines the recognition capability of our G-CSSL on the COIL-100 database. This database has 7,200 images of 100 objects. The objects were placed on a motorized turntable against a black background. The turntable was rotated through 360 degrees to vary object pose with respect to a fixed color camera. Images of the objects were taken at pose intervals of five degrees, corresponding to 72 different poses per object. We show some sample images of the database in Fig. 11. We mainly test G-CSSL for object visualization and recognition.

5.3.1 Object Visualization Analysis

We first evaluate the presented G-CSSL for visualizing the embedded data of the objects to visually perceive the low-dimensional representation. In this visualization task, six objects are selected, so there are totally 432 images. For semi-supervised learning, we choose five images from each object as labeled and the remaining as unlabeled. The visualization power of G-CSSL is compared with SSDR, SSML, MS^{3}MP, SODA and G-CSSL. We show the two-dimensional embedding result of each algorithm in Fig. 12. We can see that each method can capture the intrinsic global or local structures of object images to some extent. But note that more inter-object images are projected to be in vicinity of the embedding spaces of SSDR and SSML. Similarly, although SODA succeeds achieving the enhanced intra-object compactness, but it fails to deliver enhanced inter-object separation in the embedding space at the same time, because four different objects are projected to a close field, which may result in higher classification errors. In contrast, MS^{3}MP, G-CSSL and our TG-CSSL achieve more separated embeddings of the objects, which may contribute to booting performance.

5.3.2 Object Recognition

This study tests our TG-CSSL method for recognizing the objects. In this simulation, the first 40 objects (totally 2,880 images) of the database are selected. We prepare three settings under different numbers of labeled data (i.e., 5, 10 and 15 labeled respectively) that are randomly selected from each object class. For each setting, the number of unlabeled data is double the labeled number and the numbers of reduced dimensions are regulated from 3 to 60 with interval 3 for each fixed training size. We show the results in Fig. 13, where we also show the setting of the training set. We have the following findings. First, the increasing numbers of training data significantly boost the performance of each algorithm. Second, for each training size, the overall performance of each method goes up when the number of reduced dimensions increases in most cases. Specifically, the results of MS^{3}MP, OMS^{3}MP, G-CSSL and our TG-CSSL increase faster than those of SSDR, SSML and SODA in each case. Note that the results of MS^{3}MP, OMS^{3}MP, G-CSSL and our TG-CSSL start to go down after certain special number of d in the first case. Third, SSDR can always outperform SSML in each setting. SODA delivers comparable results to SSDR and SSML when $d<30$, but it outperforms them after $d>30$ in most cases. More specifically, SODA obtains close results to MS^{3}MP, OMS^{3}MP,

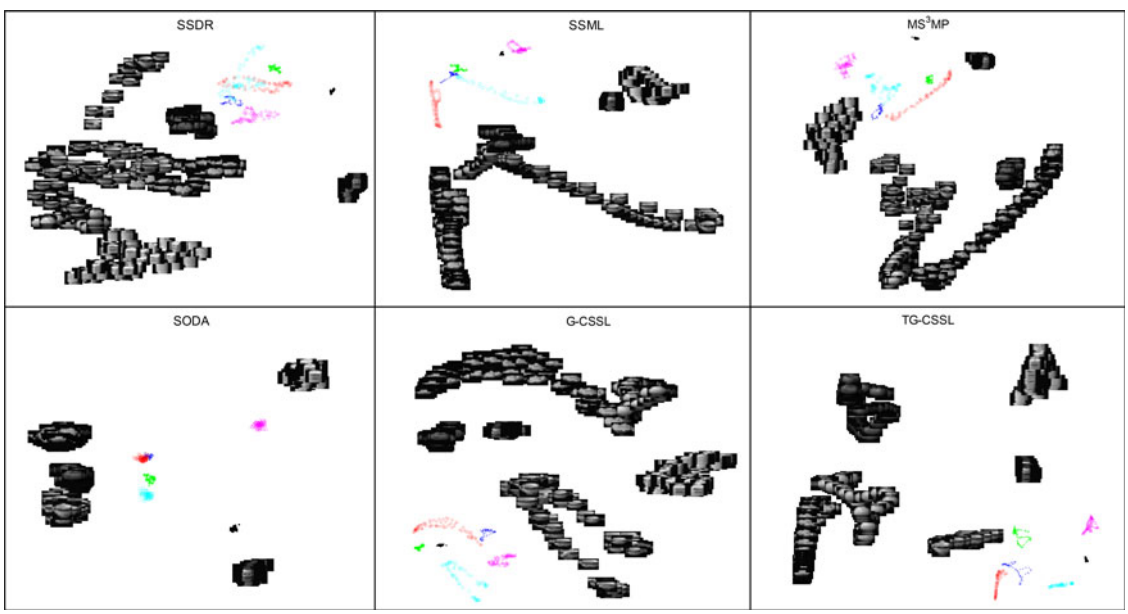


Fig. 12. Two-dimensional embedding of each algorithm on six objects of the COIL-100 database.

TABLE 3
Performance Comparison of the Algorithms on the COIL-100 Object Database

Result/Method	COIL-100 (5 labeled)			COIL-100 (10 labeled)			COIL-100 (15 labeled)		
	Mean	Best	Dim	Mean	Best	Dim	Mean	Best	Dim
SSDR	0.7298	0.7508	36	0.8306	0.8529	33	0.8741	0.9048	48
SSML	0.6956	0.7241	30	0.7983	0.8230	21	0.8272	0.8614	51
MS^{3}MP	0.8024	0.8220	24	0.8824	0.8997	27	0.9086	0.9261	30
OMS^{3}MP	0.8026	0.8147	27	0.8786	0.8926	27	0.9046	0.9173	27
SODA	0.7222	0.7870	60	0.8394	0.9069	60	0.8852	0.9399	60
SC based G-CSSL	0.8118	0.8289	12	0.8932	0.9063	27	0.9260	0.9381	54
KSVD based G-CSSL	0.8120	0.8312	15	0.8987	0.9136	30	0.9327	0.9459	51
TSC based G-CSSL	0.8191	0.8360	18	0.9015	0.9120	24	0.9328	0.9457	57

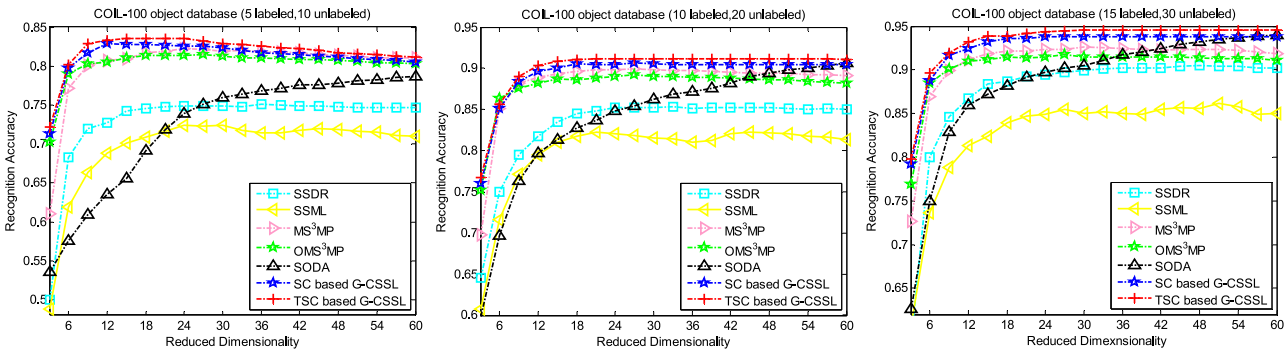


Fig. 13. Recognition accuracy versus different reduced dimensions on the COIL-100 database.

G-CSSL and TG-CSSL when more reduced dimensions are applied in most cases. MS^{3}MP and OMS^{3}MP achieve the comparative results to G-CSSL, and both are slightly worse than TG-CSSL in some cases.

Table 3 summarizes the mean accuracy and the highest accuracy according to Fig. 13 under various reduced dimensions. In Table 3, we also report the results of KSVD based G-CSSL. Similar observations can be found here, including the performance enhancement of each algorithm brought by the increased training data, and the performance superiority (i.e., mean and best records) of MS^{3}MP, OMS^{3}MP, G-CSSL, KSVD based G-CSSL and TG-CSSL over SSML and SSDR in each case. TG-CSSL delivers comparable results to KSVD based G-CSSL and both outperform other methods for this recognition task in most cases. The mean and highest accuracies of MS^{3}MP and OMS^{3}MP are always comparable to G-CSSL. Due to the fact that, in each setting with fixed training size, the performances of G-CSSL, KSVD based G-CSSL and TG-CSSL can be boosted faster by the increased number of d, they achieve more superior optimal subspace compared with other methods, since best results are delivered using smaller d values in most cases.

6 Concluding Remarks

We have discussed the insufficient data labeling problem in constrained semi-supervised learning. This paper has introduced a novel mechanism to achieve more supervised information by creating and enriching the pairwise constraint sets using the propagated soft labels through special label propagation. To improve SLP, a two-stage sparse coding approach was proposed delivering adaptive neighborhood for weight assignments. To boost the learning performance by enhancing intra-class compactness and inter-class separation, a novel voting strategy based mixed soft-similarity measure over the propagated outputs and sparse codes is also proposed. Finally, a new graph based constrained semi-supervised learning framework, termed G-CSSL, was proposed to reduce the dimensionality of data and embed new data in classification. The orthogonal projection matrix of G-CSSL can be analytically obtained using eigen-decomposition.

In this study, we mainly evaluate TSC-SLP for tansductive learning, and test G-CSSL for recognition. Although promising tansductive and recognition results have been delivered using our algorithms, the following future work is still worth exploring. First, investigating speeding up the sparse coding process with effectiveness ensured is important. Second, it is interesting to extend our G-CSSL to the practical applications, but most real observations are often corrupted by noise (that may be stochastic or deterministic) [49] and poor (or even no) alignment [50]. It is worth noting that the processes of learning dictionary and correcting errors in our framework is an independent process by PCP, so it is easy to replace the standard PCP by its extensions, such as stable PCP [49] or robust batch alignment method of linearly correlated images [50].

Acknowledgments

This work is partially supported by the National Natural Science Foundation of China (Grant Nos. 61402310 and 61373093), the Major Program of Natural Science Foundation of Jiangsu Higher Education Institutions of China (Grant No.15KJA520002), and Natural Science Foundation of Jiangsu Province of China (Grant No. BK20140008).

REFERENCES

[1] O. Chapelle, B. Scholkopf, and A. Zien, *Semi-Supervised Learning*. Cambridge, U.K.: MIT Press, 2006.

[2] G. C. Liu, Z. C. Lin, S. C. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 85, no. 1, pp. 663–670, Jan. 2012.

[3] Z. Lin, M. Chen, L. Wu, and Y. Ma, "The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices," University of Illinois at Urbana-Champaign, Champaign, IL, USA, Tech. Rep. UILU-ENG-09–2215, 2009.

[4] E. Candes, X. D. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *J. ACM*, vol. 58, no. 3, pp. 1–37, 2011.

[5] J. Mairal, F. Bach, J. Ponce, G.Sapiro, A. Zisserman, "Supervised dictionary learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2008, pp. 1033–1040.

[6] M. Aharon, M. Elad, and A. Bruckstein, "K-svd: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006.

[7] M. Zheng, J. Bu, C. Chen, C. Wang, L. J. Zhang, G. Qiu, and D. Cai, "Graph regularized sparse coding for image representation," *IEEE Trans. Image Process.*, vol. 20, no. 5, pp. 1327–1336, May 2011.

[8] S. Roweis and L. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[9] F. Wang, and C. S. Zhang, "Label propagation through linear neighborhoods," *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 1, pp. 55–67, Jan. 2008.

[10] Z. Tian, and R. Kuang, "Global linear neighborhoods for efficient label propagation," in *Proc. 12th SIAM Int. Conf. Data Mining*, pp. 863–872, 2013.

[11] C. Cortes and M. Mohri, "On transductive regression," in *Proc. Adv. Neural Inf. Process. Syst.*, 2007, pp. 305–312.

[12] M. Belkin, P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Comput.*, vol. 15, no. 6, pp. 1373–1396, 2003.

[13] N. Yang, Y. Sang, R. He, and X. Wang, "Label propagation algorithm based on non-negative sparse representation," in *Proc. Int. Conf. Life Syst. Model. Intell. Comput.*, 2010, pp. 348–357.

[14] F. Zang, and J. S. Zhang, "Label propagation through sparse neighborhood and its applications," *Neurocomputing*, vol. 97, pp. 267–277, 2012.

[15] H. Cheng, Z. Liu, and J. Yang, "Sparsity induced similarity measure for label propagation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2009, pp. 317–324.

[16] X. Zhu, Z. Ghahramani, and J. D. Lafferty, "Semi-supervised learning using Gaussian fields and harmonic functions," presented at the International Conf. Machine Learning, Washington, DC, USA, 2003.

[17] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Scholkopf, "Learning with local and global consistency," in *Proc. Adv. Neural Inf. Process. Syst.*, 2004, pp. 321–328.

[18] Y. Liu, F. P. Nie, J. G. Wu, and L. H. Chen, "Semi-supervised feature selection based on label propagation and subset selection," in *Proc. Int. Conf. Comput. Inf. Appl.*, 2010, pp. 293–296.

[19] F. P. Nie, S. M. Xiang, Y. Liu, and C. S. Zhang, "A general graph-based semi-supervised learning with novel class discovery," *Neural Comput. Appl.*, vol. 19, no. 4, pp. 549–555, 2010.

[20] L. S. Qiao, S. C. Chen, and X. Y. Tan, "Sparsity preserving projections with applications to face recognition," *Pattern Recog.*, vol. 43, no. 1, pp. 331–341, 2010.

[21] Z. Zhang, S. C. Yan, and M. B. Zhao, "Pairwise sparsity preserving embedding for unsupervised subspace learning and classification," *IEEE Trans. Image Process.*, vol. 22, no. 12, pp. 4640–4651, Dec. 2013.

[22] J. Wright, A. Yang, S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, Feb. 2009.

[23] Z. Zhang, M. B. Zhao, and T. W. S. Chow, "Binary- and multi-class group sparse canonical correlation analysis for feature extraction and classification,"*IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 10, pp. 2192–2205, Oct. 2013.

[24] P. Favaro, R. Vidal, and A. Ravichandran, "A closed form solution to robust subspace estimation and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2011, pp. 1801–1807.

[25] Y. Zhang, "Recent advances in alternating direction methods: Practice and theory," *Tutorial*, 2010.

[26] Z. Zhang, M. B. Zhao, and T. W. S. Chow, "Marginal semi-supervised sub-manifold projections with informative constraints for dimensionality reduction and recognition," *Neural Netw.*, vol. 36, pp. 97–111, 2012.

[27] E. Elhamifar, and R. Vidal, "Sparse subspace clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2009, pp. 2790–2797.

[28] D. Q. Zhang, Z. H. Zhou, and S. C. Chen, "Semi-supervised dimensionality reduction," in *Proc. SIAM Int. Conf. Data Mining*, Minneapolis, MN, USA, 2007, pp. 11–393.

[29] M. S. Baghshah and S. B. Shouraki, "Semi-supervised metric learning using pairwise constraints," in *Proc. Int. Joint Conf. Artif. Intell.*, 2009, pp. 1217–1222.

[30] Z. Zhang, T. W. S. Chow, and M.B. Zhao, "M-Isomap: Orthogonal constrained marginal isomap for nonlinear dimensionality reduction," *IEEE Trans. Syst., Man Cybern. B, Cybern.*, vol. 43, no. 1, pp. 180–192, Feb. 2013.

[31] Z. Zhang, T. W. S. Chow, and M. B. Zhao, "Trace ratio optimization based semi-supervised nonlinear dimensionality reduction for marginal manifold visualization," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 5, pp. 1148–1161, May 2013.

[32] M. Wang, X. Hua, T. Mei, R. Hong, G. Qi, Y. Song, and L. Dai, "Semi-supervised kernel density estimation for video annotation," *Comput. Vis. Image Understanding*, vol. 13, no. 3, pp. 384–396, 2009.

[33] J. Chen, J. Ye, and Q. Li, "Integrating global and local structures: A least squares framework for dimensionality reduction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2007, pp. 1–8.

[34] X. Zhu, "Semi-supervised learning literature survey," Univ. Wisconsin-Madison, Madison, WI, USA, Tech. Rep. 1530, 2005.

[35] Z. Zhang, T. W. S. Chow, and N. Ye, "Semi-supervised multimodal dimensionality reduction,"*Comput. Intell.*, vol. 29, no. 1, pp. 70–110, 2013.

[36] D. Cai, X. F. He, and J.W. Han, "Semi-supervised discriminant analysis," in *Proc. IEEE Int. Conf. Comput. Vis.*, Rio de Janeiro, Brazil, 2007, pp. 1–7.

[37] Y. Q. Song, F. P. Nie, C. S. Zhang, and S. M. Xiang, "A unified framework for semi-supervised dimensionality reduction," *Pattern Recog.*, vol. 41, no. 9, pp. 2789–2799, 2008.

[38] H. Li, T. Jiang and K. Zhang, "Efficient and robust feature extraction by maximum margin criterion," *IEEE Trans. Neural Netw.*, vol. 17, no. 1, pp. 157–165, Jan. 2006.

[39] Z. Zhang, and N. Ye, "Learning a tensor subspace for semi-supervised dimensionality reduction," *Soft Comput.*, vol. 15, no. 2, pp. 383–395, 2011.

[40] F. P. Nie, S. M. Xiang, Y. Q. Jia, and C. S. Zhang, "Semi-Supervised orthogonal discriminant analysis via label propagation," *Pattern Recog.*, vol. 42, no. 11, pp. 2615–2627, 2009.

[41] E. Kokiopoulou and Y. Saad, "Orthogonal neighborhood preserving projections: A projection-based dimensionality reduction technique," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 12, pp. 2143–2156, Dec. 2007.

[42] J. Cai, E. Candes, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM J. Optimization*, vol. 20, no. 4, pp. 1956–1982, 2010.

[43] M. B. Zhao, Z. Zhang, and W. S. Chow, "Trace ratio criterion based generalized discriminative learning for semi-supervised dimensionality reduction," *Pattern Recog.*, vol. 45, no. 4, pp. 1482–1499, 2012.

[44] Y. Ma, J. Wright, Z. C. Lin, and A. Y. Yang, "The pursuit of low-dimensional structures in high-dimensional data," *Tutorial*, 2010.

[45] I. Jolliffe, *Principal Component Analysis*. New York, NY, USA: Springer-Verlag, 1986.

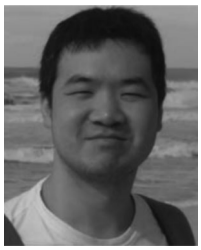
[46] J. Yang, A. F. Frangi, J. Y. Yang, D. Zhang, and Z. Jin, "KPCA plus LDA: A complete kernel Fisher discriminant framework for feature extraction and recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 2, pp. 234–244, Feb. 2005.

[47] Z. Zhang, M. B. Zhao, and T. W. S. Chow, "Label propagation and soft-similarity measure for graph based constrained semi-supervised learning," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, Beijing, China, 2014, pp. 2927–2934.

[48] J. Wright, Y. Y. Tao, Z. C. Lin, Y. Ma, and H. Y. Shum, "Classification via minimum incremental coding length (MICL)," in *Proc. Adv. Neural Inf. Process. Syst.*, 2008, pp. 1633–1640.

[49] Z. H. Zhou, X. D. Li, J. Wright, E. J. Candès, and Y. Ma, "Stable principal component pursuit," in *Proc. IEEE Int. Symp. Inf. Theory*, 2010, pp. 1518–1522.
[50] Y. G. Peng, A. Ganesh, J. Wright, W. L. Xu, and Y. Ma, "RASL: Robust alignment by sparse and low-rank decomposition for linearly correlated images,"*IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2233–2246, Nov. 2012.

Zhao Zhang (M'13) received the PhD degree in computer engineering from the Department of Electronic Engineering (EE), City University of Hong Kong, where advised by Prof. Tommy W. S. Chow, in 2013. Dr. Zhang is now an associate professor in the School of Computer Science and Technology, Soochow University, Suzhou, China. He was a visiting research engineer at the Learning &Vision Research Group, National University of Singapore, where he worked with Prof. Shuicheng Yan (IAPR Fellow), from Feb. to May 2012. He then visited the National Laboratory of Pattern Recognition (NLPR) at Chinese Academy of Sciences (CAS), where he worked with Prof. Cheng-Lin Liu (*Director of the Laboratory, IEEE Fellow, IAPR Fellow*), from Sept. to Dec. 2012. His current research interests include pattern recognition, machine learning & data mining, and computer vision. He has authored/co-authored more than 30 technical papers published at prestigious international journals and conferences, including *IEEE TIP, IEEE TKDE, IEEE TSMC-B, ACM TIST, Pattern Recognition (PR), Neural Networks (NN)*, ACM ICMR, and ICPR, etc. Dr. Zhang is now serving on the editorial board of the *Journal of Pattern Recognition Research*. He served as the Area Chair (AC) for BMVC 2015, and also served as a Program Committee (PC) member for several important international conferences, including SDM 2015, IJCNN 2015, CAIP 2015, etc. He is now a member of the IEEE, and a member of the China Computer Federation (CCF).



Mingbo Zhao (S'11-M'13) received the BSc and master's degrees from the Department of Electronic Engineering, Shanxi University, Shanxi, China, in 2005 and 2008, respectively. He received the PhD degree in computer engineering from the Department of Electronic Engineering, City University of Hong Kong, in 2013. He is currently a senior research assistant in the Department of Electronic Engineering, City University of Hong Kong. His interests include machine learning, data mining, and pattern recognition. He is a member of the IEEE.

Tommy W.S. Chow (M'93-SM'03) received the BSc (First Hons.) and PhD degrees from the University of Sunderland, Sunderland, UK. He is currently a full professor in the Electronic Engineering Department, City University of Hong Kong, Hong Kong SAR. He has been an active Committee Member of HKIE (Hong Kong Institution of Engineers) Control Automation and Instrumentation (CAI) division since 1992, and was the Division Chairman (1997-1998) for HKIE CAI division. Prof. Chow received the best paper award from the 28th Annual Conference of the IEEE Industrial Electronics Society (IECON 2002). He has authored or coauthored over 160 journals articles, 5 book chapters, and over 60 conference papers. He is now serving as an associate editor for the *IEEE Transactions on Industrial Informatics*. He is a senior member of the IEEE.