Learning Semi-Supervised Representation Towards a Unified Optimization Framework for Semi-Supervised Learning

Chun-Guang Li[1], Zhouchen Lin[2,3], Honggang Zhang[1], and Jun Guo[1]

[1] School of Info. & Commu. Engineering, Beijing University of Posts and Telecommunications
[2] Key Laboratory of Machine Perception (MOE), School of EECS, Peking University
[3] Cooperative Medianet Innovation Center, Shanghai Jiaotong University

{lichunguang, zhhg, guojun}@bupt.edu.cn; zlin@pku.edu.cn

Abstract

State of the art approaches for Semi-Supervised Learning (SSL) usually follow a two-stage framework – constructing an affinity matrix from the data and then propagating the partial labels on this affinity matrix to infer those unknown labels. While such a two-stage framework has been successful in many applications, solving two subproblems separately only once is still suboptimal because it does not fully exploit the correlation between the affinity and the labels. In this paper, we formulate the two stages of SSL into a unified optimization framework, which learns both the affinity matrix and the unknown labels simultaneously. In the unified framework, both the given labels and the estimated labels are used to learn the affinity matrix and to infer the unknown labels. We solve the unified optimization problem via an alternating direction method of multipliers combined with label propagation. Extensive experiments on a synthetic data set and several benchmark data sets demonstrate the effectiveness of our approach.

1. Introduction

In real world applications, one often faces the scenario that the acquisition of data with labels is quite costly in human labor, whereas a large amount of unlabeled data are relatively easy to obtain. Therefore, Semi-Supervised Learning (SSL) was proposed to incorporate both unlabeled data and labeled data for classification [26, 6, 7, 33].

1.1. Related Work

In the past decade, graph-based SSL approaches have attracted much attention for its simple and elegant formulation and a number of algorithms have been proposed, e.g., [26, 34, 17, 31, 4, 18, 28, 19, 11, 30, 15]. For a more detailed survey, please refer to [33].

The central idea behind the graph-based SSL approaches is to explore the pairwise affinity between data points to infer those unknown labels. More concretely, the unknown labels should be consistent with both the given partial labels and the pairwise affinity. In this sense, graph-based SSL approaches are usually interpreted as a process of propagating labels through the pairwise affinity of data points, which is called *label propagation* [31, 33, 8].

The affinity to measure the similarity of data points and the mechanism to infer those unknown labels are two fundamental components in SSL. Roughly speaking, different approaches differ in the way to induce the affinity matrix, e.g., [34, 31, 32, 28, 11, 30, 10, 15, 36], and/or the mechanics to infer those unknown labels, e.g., [26, 34, 17, 31, 32, 16, 1, 28, 19].

The existing approaches to define the affinity can be roughly divided into three categories: a) the Euclidean distance based approaches, which use the k nearest neighbors rule to find the local neighborhood and encode the proximity among data points as binary weights or as "soft" weights by a heat kernel, e.g., [5, 34, 6, 12]; b) the local self-expressiveness model based approaches, which induce the affinity by expressing each data point as a linear combination of local neighbors, e.g., [25, 28]; c) the global self-expressiveness model based approaches, which induce the affinity by expressing each data point as a linear combination of all other data points, e.g., sparsity induced affinity [11, 30, 15, 36] and low-rankness induced affinity [23]. The Euclidean distance based approaches and the local self-expressiveness model based approaches depend upon the local neighborhood parameter (e.g., k) and there is no reliable approach to determine its optimal value.

While the mechanics to infer those unknown labels being

interpreted as a process of propagating labels through the pairwise affinity, their implementation strategies are different, e.g., Markov random walks [26], the harmonic function approach [34], spectral graph transducer [17], the local and global consistency approach [31], the Green's function approach [32], the online approach [16], the Gaussian process [1], and the robust label propagation [19].

Recent advances in SSL are more emphasizing on constructing an informative affinity matrix [28, 11, 30, 10, 29, 15, 36]. Although these approaches have been very successful in many applications, an important shortcoming is that they divide the SSL problem into two separate stages:

- Construct the affinity matrix, using, e.g., heat kernel [6], local linear representation [25, 28], sparse representation [11, 30, 10, 15], and low-rank representation [23].
- Infer the unknown labels, using, e.g., label propagation via harmonic function [34], and the local and global consistency approach [31].

While the two subproblems can be easily solved separately, the major disadvantage is that the natural relationship between the affinity matrix and the labels of the data is not fully exploited. Notice that in the approaches mentioned above, the affinity matrix is induced by using *neither* the given labels *nor* the estimated labels, and the label propagation is conducted by using the given labels only. A very interesting work is [35], in which the given partial labels are used to construct a kernel and then a transductive support vector machine is adopted to estimate unknown labels. While the value of the partial labels for inducing a better affinity matrix was verified, neither the estimated labels were exploited nor a unified single objective was built.

In this paper, we attempt to integrate the two separate stages into a single unified optimization framework. One important observation is that the estimated labels of the data can provide useful "weakly" supervised information for building a better affinity matrix and facilitate label propagation. Because of this, if we feed back the "weakly" supervised information induced from the estimated labels properly, it is possible to yield a better affinity matrix and more accurate estimation of the unknown labels.

Paper Contributions. We propose to integrate the separate two stages of SSL into a unified optimization framework, in which both the given labels and the estimated labels are used to learn the affinity matrix and to infer the unknown labels. Different from the previous two-stage approaches, our unified optimization framework fully exploits both the given labels and the estimated labels. To the best of our knowledge, this is the first attempt to build a unified optimization framework for SSL to fully exploit the given labels and the inferred labels to revise the affinity matrix and to facilitate the label propagation. In experiments, we will show that the classification accuracy could be boosted significantly during the iterations.

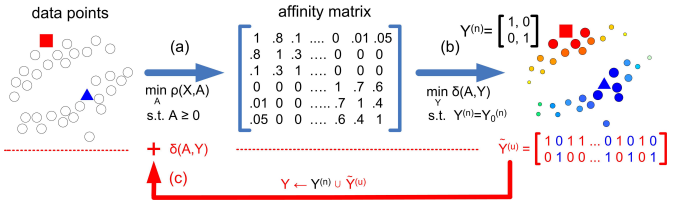


Figure 1. Illustrations of the "single pass" two-stage SSL paradigm and the proposed unified optimization framework. (a) Construct an affinity matrix. (b) Infer the unknown labels. (c) Incorporate the given and inferred labels to revise the affinity matrix and to facilitate the label propagation.

2. A Unified Optimization Framework for Semi-Supervised Learning

This paper considers with the following problem.

Problem 2.1 (**Semi-Supervised Learning**). *Given a data set* $\mathcal{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n, \boldsymbol{x}_{n+1}, \ldots, \boldsymbol{x}_N\}$ *which consists of* m *classes, where data point* $\boldsymbol{x}_j \in \mathbb{R}^d$*, and a few labels* $\mathcal{Y}_n = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n\}$*, where the label* $\boldsymbol{y}_j$ *is an* m*-dimensional indicator vector (in which* $\boldsymbol{y}_{ij} = 1$ *corresponds to the* i*-th class),* n *is the number of labeled data points,* $N - n$ *is the number of unlabeled data points, and* $n \ll N$*. The goal of SSL is to estimate the unknown labels for the remaindering* $N - n$ *data points.*

Without loss of generality, we arrange the labeled data points as the columns of matrix $X^{(n)} = \left[\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\right]$, the unlabeled data points as the columns of matrix $X^{(u)} = \left[\boldsymbol{x}_{n+1}, \boldsymbol{x}_{n+2}, \ldots, \boldsymbol{x}_{n+u}\right]$, where $N = n + u$, and hence we have $X = \left[X^{(n)}, X^{(u)}\right]$. Accordingly, we divide the label matrix Y as $Y = \left[Y^{(n)}, Y^{(u)}\right]$, in which $Y^{(n)} = \left[\boldsymbol{y}_1, \cdots, \boldsymbol{y}_n\right]$ is the known label matrix and $Y^{(u)} = \left[\boldsymbol{y}_{n+1}, \cdots, \boldsymbol{y}_N\right]$ is the unknown label matrix. Since each data point lies in only one class, we have $Y^\top \mathbf{1} = \mathbf{1}$, where $\mathbf{1}$ is the vector of all 1's of appropriate dimension. Notice also that the number of classes is equal to m and thus we have $\text{rank}(Y) = m$. Consequently, we define the space of label matrices as

$$\mathcal{Y} = \{Y \in \{0,1\}^{m \times N} : Y^\top \mathbf{1} = \mathbf{1},\ \text{rank}(Y) = m\}. \quad (1)$$

Note that in the setting of SSL, what we need to do is to construct an affinity matrix and then estimate the unknown part $Y^{(u)}$ based on a few labels in $Y^{(n)}$.

2.1. Reformulating Semi-Supervised Learning

Recall that existing approaches [33, 28, 11, 30] for SSL usually divide the problem into two separate stages: com-

puting an affinity matrix and then inferring the unknown labels. For the affinity matrix, we expect that the connections among data points from different classes are as weak (or sparse) as possible; whereas the connections among data points within each class are as strong (or dense) as possible. Ideally, the sufficient condition for perfect classification is that the affinity matrix should be block-diagonal in which each block is connected and corresponds to data points from the same class.

At this moment, suppose that the label matrix Y *was* known. An interesting fact is that the block-diagonal patterns of an ideal affinity matrix A and the labels in matrix Y are related. Notice that ideally whenever data points i and j belong to different classes, we must have $A_{ij} = 0$. In another words, we have $A_{ij} \neq 0$ if and only if data points i and j are in the same class, and thus we must have $\boldsymbol{y}_i = \boldsymbol{y}_j$ where $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$ are the i-th and the j-th columns of matrix Y, respectively. Therefore, we can quantify the disagreement between the affinity matrix A and the label matrix Y by using the following measure:

$$\begin{aligned} \delta(A, Y) &= \sum_{i,j} |A_{ij}|(\frac{1}{2}\|\boldsymbol{y}_i - \boldsymbol{y}_j\|^2) \\ &= \|\Theta \odot A\|_1, \end{aligned} \tag{2}$$

where $\Theta_{ij} = \frac{1}{2}\|\boldsymbol{y}_i - \boldsymbol{y}_j\|^2$, the operator $\odot$ indicates the element-wise product, and $\|\cdot\|_1$ is the vector ℓ_1 norm. We call $\delta(A, Y)$ as a disagreement measure of A with respect to (w.r.t.) the label matrix Y. Since $\Theta_{ij} \in \{0, 1\}$, the disagreement measure $\delta(A, Y)$ in (2) effectively counts the times that A and Y disagree and weights this count by $|A_{ij}|$. Note that in the ideal case – that is, the affinity matrix A is block-diagonal and each block corresponds to a class, i.e., $A_{ij} = 0$ whenever points i and j are in different classes, then the disagreement measure vanishes; otherwise it is positive.

In the practical setting of SSL, the label matrix Y is not completely known. Nevertheless, the disagreement measure $\delta(A, Y)$ still provides a useful measure for the inconsistency between the affinity matrix A and the estimated labels in Y. Recall that the main idea of graph-based SSL approaches is to estimate the unknown labels that are consistent with the pairwise affinities and the given initial labels whenever possible [34, 8]. Therefore, those unknown labels $Y^{(u)}$ can be estimated by minimizing the disagreement measure $\delta(A, Y)$ over Y subject to the given labels in $Y^{(n)}$. Precisely, we summarize the two-stage paradigm of the existing SSL as follows:

- Step 1. Computing the affinity matrix A by solving

$$\min_{A} \ \rho(X, A) \ \text{ s.t. } \ A \geq 0, \tag{3}$$

  where $\rho(\cdot, \cdot)$ is an implicit or explicit function which corresponds to the specific approach for constructing the affinity matrix A based on data matrix X, and the constraint $A \geq 0$ is for guaranteeing the nonnegativity of the entries in A.

- Step 2. Estimating the label matrix $Y^{(u)}$ by solving

$$\min_{Y} \ \delta(A, Y) \ \text{ s.t. } \ Y^{(n)} = Y_0^{(n)}, \ \ Y \in \mathcal{Y}, \tag{4}$$

  where $Y = \left[Y^{(n)}, Y^{(u)}\right]$ is the label matrix in which $Y^{(n)}$ is given but $Y^{(u)}$ is unknown.

We illustrate the "single pass" two-stage procedure for SSL in Fig. 1 (a) and (b). Notice that the two-stage paradigm is sub-optimal, because dividing an SSL problem into two separate steps and solving each of them individually in a single pass do not fully exploit the correlation between the affinity matrix and the labels. To be more specific, the existing approaches for computing the affinity matrix A as in (3) exploit *neither* the given labels in $Y^{(n)}$ *nor* the estimated labels in $Y^{(u)}$, and the existing approaches for inferring the unknown labels exploit only the given labels.

2.2. Building a Unified Framework for Semi-Supervised Learning

In this paper, we formulate the SSL problem into a unified optimization framework over the affinity matrix A and the space of label matrices $\mathcal{Y}$ simultaneously as follows:

$$\begin{aligned} &\min_{A,Y} \ \rho(X, A) + \gamma\delta(A, Y) \\ &\text{s.t. } \ A \geq 0, \ \ Y^{(n)} = Y_0^{(n)}, \ \ Y \in \mathcal{Y}, \end{aligned} \tag{5}$$

where $\gamma > 0$ is a tradeoff parameter. As illustrated in Fig. 1, by building a feedback path as in (c), the given partial labels and the estimated labels can be automatically incorporated to induce the affinity matrix and facilitate the label propagation in the next iteration.

The proposed optimization framework in (5) generalizes the existing SSL paradigm because, instead of first solving for the affinity matrix A and then applying, e.g., label propagation techniques [34, 31], to obtain the unknown labels, we simultaneously search for the affinity matrix A and the unknown labels $Y^{(u)}$. In practice, the problem in (5) can be solved alternately as below:

- Given A and $Y^{(n)}$, the problem for solving $Y^{(u)}$ is a standard label propagation problem in SSL.

- Given Y, the problem is to incorporate the supervised and weakly supervised information in Y to update the affinity matrix A.

Solving Y when A and $Y^{(n)}$ are given. Note that $\delta(A, Y)$ can be rewritten as

$$\delta(A, Y) = \sum_{i,j} |A_{ij}|(\frac{1}{2}\|\boldsymbol{y}_i - \boldsymbol{y}_j\|^2) = \text{trace}(Y\bar{L}Y^\top), \tag{6}$$

where $\bar{L} = \bar{D} - \bar{A}$ is the graph Laplacian, $\bar{A} = \frac{1}{2}(A + A^\top)$ and $\bar{D}$ is a diagonal matrix whose diagonal entries are $\bar{D}_{jj} = \sum_i \bar{A}_{ij}$. Then given A we can find Y by solving the problem

$$\min_{Y} \operatorname{trace}(Y\bar{L}Y^\top) \quad \text{s.t.} \quad Y^{(n)} = Y_0^{(n)}, \quad Y \in \mathcal{Y}, \tag{7}$$

which is the problem solved approximately by label propagation approaches [34, 31].

Recomputing A when Y is updated. Recall that the state of the art approaches induce the affinity matrix A via the coefficients of a *self-expressiveness model* by using, e.g., $A = \frac{1}{2}(|C| + |C^\top|)$ where C is the coefficients matrix, in which each data point is expressed as a linear combination of other data points, i.e., $X = XC$, e.g., LLR [25, 28], SR [30, 11, 13], LRR [23]. Formally, the state of the art self-expressiveness models, e.g., SR [30, 11, 13], LRR [23, 22], compute the coefficients matrix C by solving the following problem:

$$\min_{C,E} \|C\|_\kappa + \lambda\|E\|_\ell \text{ s.t. } X = XC + E, \ \operatorname{diag}(C) = 0, \tag{8}$$

where $\|\cdot\|_\kappa$ and $\|\cdot\|_\ell$ are two properly chosen norms on C and E, respectively, $\operatorname{diag}(C) = 0$ is an optional constraint to rule out the trivial solution, and $\lambda > 0$ is a tradeoff parameter. This process virtually addresses problem (3), that is, the implicit functional $\rho(X, A)$ corresponds to solving problem (8) at first and then defining the affinity matrix A via $\frac{1}{2}(|C_*| + |C_*^\top|)$, where C_* is the optimal solution of C.

Notice that in our unified optimization framework (5), when Y is given, the affinity matrix A is induced by solving the following problem:

$$\min_{A} \rho(X, A) + \gamma\delta(A, Y) \ \text{ s.t. } \ A \geq 0. \tag{9}$$

Note that not only the partial supervision information in label matrix $Y^{(n)}$ but also the estimated label matrix $Y^{(u)}$ can be automatically incorporated in computing or revising the affinity matrix A. By substituting the self-expressiveness model (8) into problem (9), we have a *semi-supervised self-expressiveness* model:

$$\begin{aligned} &\min_{C,E} \ \|C\|_\kappa + \gamma\|\Theta \odot C\|_1 + \lambda\|E\|_\ell \\ &\text{s.t. } \ X = XC + E, \ \ \operatorname{diag}(C) = 0, \end{aligned} \tag{10}$$

where $\delta(A, Y)$ is replaced by $\|\Theta \odot C\|_1$. The partial supervision information in given labels $Y^{(n)}$ and the weakly supervised information in the estimated labels $Y^{(u)}$ is encoded in the semi-supervised structure matrix Θ.[1]

When using the nuclear norm $\|\cdot\|_*$ to replace $\|\cdot\|_\kappa$, we have a Semi-Supervised Low-Rank Representation (S^{2}LRR) which solves the following problem:

$$\begin{aligned} &\min_{C,E} \ \|C\|_* + \gamma\|\Theta \odot C\|_1 + \lambda\|E\|_\ell \\ &\text{s.t. } \ X = XC + E, \ \ \operatorname{diag}(C) = 0, \end{aligned} \tag{11}$$

where the norm $\|\cdot\|_\ell$ on the error term E depends upon the prior knowledge about the pattern of noise or corruptions.

When using the ℓ_1 norm $\|\cdot\|_1$ to replace $\|\cdot\|_\kappa$, we obtain a Semi-Supervised Sparse Representation (S^{3}R) which solves the following problem:

$$\begin{aligned} &\min_{C,E} \ \|C\|_1 + \gamma\|\Theta \odot C\|_1 + \lambda\|E\|_\ell \\ &\text{s.t. } \ X = XC + E, \ \ \operatorname{diag}(C) = 0. \end{aligned} \tag{12}$$

By employing S^{2}LRR or S^{3}R to induce the affinity matrix, the unified optimization framework in (5) is implemented as follows:

$$\begin{aligned} &\min_{C,E,Y} \|C\|_\kappa + \gamma\|\Theta \odot C\|_1 + \lambda\|E\|_\ell \\ &\text{s.t. } X{=}XC{+}E, \ \operatorname{diag}(C) = 0, \ Y^{(n)}{=}Y_0^{(n)}, \ Y{\in}\mathcal{Y}, \end{aligned} \tag{13}$$

where $\|C\|_\kappa$ is a properly chosen norm, e.g., $\|C\|_*$ or $\|C\|_1$.

Remark 1. Note that when only a few labels are available, inferring labels over an imperfect affinity matrix via label propagation is not a well-posed problem, which is sensitive to several factors, e.g., the quality of the affinity matrix and the number of partial labels. In our unified model, the given partial labels and the estimated labels are combined together to refine the affinity matrix during the iterations. The better the affinity matrix is, the more stable and accurate the estimation of those unknown labels is. On the other hand, in our unified model, both the given partial labels and the estimated labels are available to infer the unknown labels in the next iteration. On average, the more the given labels are, the more stable and accurate the inferring of the unknown labels is. Because of the feedback of inferred labels to induce the affinity matrix and to facilitate label propagation, we call the unified optimization framework (13) as Self-Taught Semi-Supervised Learning (STSSL).

Remark 2. While there exist other variants of self-expressiveness model, e.g., nonnegative local linear representation [27, 20], nonnegative sparse representation [15], and nonnegative low-rank and sparse representation [36], we extend only the models in [13, 23] because, the nonnegative constraint can easily be adopted in our model (13).

3. Alternating Minimization Algorithm for Solving the Unified Optimization Problem

We propose to solve the optimization problem in (13) by solving the following two subproblems alternately:

1. Find C and E given Y by solving a semi-supervised low-rank or sparse representation problem.

[1] As shown in Section 3.2 that, the nonnegativity constraints over the entries of A are implicitly guaranteed.

2. Find Y given C and E by label propagation.

3.1. Semi-Supervised Representations

Solving S^{2}LRR. Given the label matrix Y (or equivalently the semi-supervision matrix Θ), we solve for C and E from structured low-rank representation problem (11) by solving an equivalent problem as follows:

$$\begin{aligned} &\min_{C,E} \ \|Z\|_* + \gamma\|\Theta \odot C\|_1 + \lambda\|E\|_\ell \\ &\text{s.t. } \ X = XZ + E, \ \ Z = C - \text{diag}(C). \end{aligned} \tag{14}$$

We solve this problem using the Linearized Alternating Direction Method of Multipliers (LADMM) [21]. The augmented Lagrangian is given by:

$$\begin{aligned} &\mathcal{L}(C, Z, E, \Lambda^{(1)}, \Lambda^{(2)}) \\ &= \|Z\|_* + \gamma\|\Theta \odot C\|_1 + \lambda\|E\|_\ell + \langle \Lambda^{(1)}, X - XZ - E\rangle \\ &+ \langle \Lambda^{(2)}, Z - C + \text{diag}(C)\rangle \\ &+ \frac{\mu}{2}(\|X - XZ - E\|_F^2 + \|Z - C + \text{diag}(C)\|_F^2), \end{aligned}$$

where $\Lambda^{(1)}$ and $\Lambda^{(2)}$ are matrices of Lagrange multipliers, and $\mu > 0$ is a penalty parameter. To find a saddle point for $\mathcal{L}$, we update each of C, Z, E, $\Lambda^{(1)}$, and $\Lambda^{(2)}$ alternately while keeping the other variables fixed.

1. Update Z by solving the following problem

$$Z_{t+1} = \arg\min_Z \frac{1}{\mu_t \eta}\|Z\|_* + \frac{1}{2}\|Z - W_t\|_F^2, \tag{15}$$

whose solution is given by the Singular Value Thresholding operator [9], i.e.,

$$Z_{t+1} = U\mathcal{S}_{\frac{1}{\mu_t\eta}}(S)V^\top, \tag{16}$$

where $\eta > \sigma_{\max}(X^\top X + 1)$, $W_t = Z_t - \frac{1}{\eta}[\frac{1}{\mu_t}(\Lambda_t^2 - X^\top\Lambda_t^{(1)}) + X^\top(XZ_t + E_t - X) + (Z_t - C_t + \text{diag}(C_t))]$, and $W_t = USV^\top$ is a skinny Singular Value Decomposition (SVD).

2. Update C by solving the following problem

$$C_{t+1} = \arg\min_C \frac{\gamma}{\mu_t}\|\Theta \odot C\|_1 + \frac{1}{2}\|C - \text{diag}(C) - A_t\|_F^2,$$

where $A_t = Z_t + \frac{1}{\mu_t}\Lambda_t^{(2)}$. The closed-form solution of C is given as

$$C_{t+1} = \tilde{C}_{t+1} - \text{diag}(\tilde{C}_{t+1}), \tag{17}$$

where the (i, j) entry of $\tilde{C}$ is given by $\tilde{C}_{t+1}^{ij} = \mathcal{S}_{\frac{\gamma\Theta_{ij}}{\mu_t}}(A_t^{ij})$, in which $\mathcal{S}_\tau(\cdot)$ is the shrinkage thresholding operator [3].

Algorithm 1 ADMM for solving problem (11) and (12)

Input: Data matrix X, label matrix Y, λ, and γ.
  Initialize: C, Z, E, $\Lambda^{(1)}$, and $\Lambda^{(2)}$, $\epsilon = 10^{-6}$, $\rho_1 = 1.1$
  while not converged **do**
    Update Z_t, C_t, and E_t;
    Update $\Lambda_t^{(1)}$ and $\Lambda_t^{(2)}$;
    Update $\mu_{t+1} = \rho\mu_t$;
    If not converged, then set $t \leftarrow t + 1$.
  end while
Output: C_{t+1} and E_{t+1}

3. Update E as follows:

$$E_{t+1} = \arg\min_E \frac{\lambda}{\mu_t}\|E\|_\ell + \frac{1}{2}\|E - V_t\|_F^2 \tag{18}$$

where $V_t = X - XZ_{t+1} + \frac{1}{\mu_t}\Lambda_t^{(1)}$. If we use the ℓ_1 norm for E, then $E_{t+1} = \mathcal{S}_{\frac{\lambda}{\mu_t}}(V_t)$.

4. Update $\Lambda^{(1)}$ and $\Lambda^{(2)}$ by:

$$\begin{aligned} \Lambda_{t+1}^{(1)} &= \Lambda_t^{(1)} + \mu_t(X - XZ_{t+1} - E_{t+1}), \\ \Lambda_{t+1}^{(2)} &= \Lambda_t^{(2)} + \mu_t(Z_{t+1} - C_{t+1} + \text{diag}(C_{t+1})). \end{aligned} \tag{19}$$

Solving S^{3}R. Given the label matrix Y (or equivalently the semi-supervision matrix Θ), we solve for C and E from structured sparse representation problem (12) by solving an equivalent problem as following

$$\begin{aligned} &\min_{C,E} \ \|C\|_1 + \gamma\|\Theta \odot C\|_1 + \lambda\|E\|_\ell \\ &\text{s.t. } \ X = XZ + E, \ \ Z = C - \text{diag}(C). \end{aligned} \tag{20}$$

Similarly, we can solve the structured sparse representation problem via ADMM with the minor changes in the steps of updating C and Z.

1. Update C by $C_{t+1} = \tilde{C}_{t+1} - \text{diag}(\tilde{C}_{t+1})$, where $\tilde{C}_{t+1}^{ij} = \mathcal{S}_{\frac{1}{\mu_t}(\gamma\Theta_{ij}+1)}(U_t^{ij})$.

2. Update Z by $Z_{t+1} = (X^\top X + \mathbf{I})^{-1}[X^\top(X - E_t - \frac{1}{\mu_t}\Lambda_t^{(1)}) + C_t - \text{diag}(C_t) - \frac{1}{\mu_t}\Lambda_t^{(2)}]$.

For clarity, we summarize the ADMM algorithm for solving problems (11) and (12) in Algorithm 1. For the details of the derivation, we refer the readers to [21].

3.2. Label Propagation

Given C and E, problem (13) reduces to the following problem:

$$\min_Y \ \|\Theta \odot C\|_1 \ \text{ s.t. } \ \ Y^{(n)} = Y_0^{(n)}, \ \ \ Y \in \mathcal{Y}, \tag{21}$$

where $\|\Theta \odot C\|_1$ is the $\delta(C, Y)$. We will show that this problem is equivalent to problem (6) by properly inducing the affinity matrix A from C. Recall that

$$\|\Theta \odot C\|_1 = \frac{1}{2}\sum_{i,j} A_{i,j}\|\boldsymbol{y}_i - \boldsymbol{y}_j\|_2^2 = \text{trace}(Y\bar{L}Y^\top), \quad (22)$$

where $A_{i,j} = \frac{1}{2}(|C_{ij}| + |C_{ji}|)$ measures the similarity of points i and j, $\bar{L} = \bar{D} - A$ is a graph Laplacian, and $\bar{D}$ is a diagonal matrix whose diagonal entries are $\bar{D}_{jj} = \sum_i A_{ij}$. Consequently, problem (21) turns out to be the following problem:

$$\min_{Y} \text{ trace}(Y\bar{L}Y^\top) \text{ s.t. } \quad Y^{(n)} = Y_0^{(n)}, \quad Y \in \mathcal{Y}. \quad (23)$$

This problem can be solved approximately with label propagation approaches, e.g., the harmonic function approach [34].

Remark 3. Notice that from the second iteration, the estimated labels are also available. To boost the accuracy of label propagation, we select a small proportion of the estimated labels to augment the set of the given labels and perform the label propagation for only the remaining data points in the following iterations. The selected labels, called "seed labels", are automatically determined during the iterations. The weakly supervised structure matrix Θ is formed by using the given partial labels in the first iteration; whereas in the second iteration the weakly supervised structure matrix Θ is formed by using the given partial labels, the selected seed labels, and the remaining inferred labels. Both the initial labels and the seed labels are used to not only revise the affinity matrix but also supervise the label propagation in the next iteration; whereas the remaining inferred labels are used to revise the affinity matrix only.

3.3. Algorithm Summary

For clarity, we summarize the alternating scheme to solve the unified optimization framework for SSL (13) in Algorithm 2 and term it STSSL. The algorithm alternates between solving (C, E) given the label matrix Y using Algorithm 1, and solving for Y given (C, E) using the label propagation approach. While the problem solved by Algorithm 1 is still a convex program, there is no guarantee that Algorithm 2 will converge to a global or local optimum because the solution for Y given (C, E) is obtained in an approximate manner. Nonetheless, our experiments show that the algorithm does converge in practice for appropriate settings of the parameters.

Stopping Criterion. Algorithm 2 can be stopped by setting a maximum iteration number $T_{\max}$ or by checking the following condition

$$\|Y_{T+1}^{(u)} - Y_T^{(u)}\|_\infty < 1, \quad (24)$$

where $T = 1, 2, \ldots,$ is the iteration number.

Algorithm 2 STSSL

Input: Data matrix X and partial label matrix $Y^{(n)}$
  Initialize: $Y^{(u)} = \mathbf{0}$, γ, and λ
  while not converged **do**
    Given Y, solve problem (11) or (12) via Algorithm 1 to obtain (C, E);
    Given (C, E), solve problem (23) to estimate $Y^{(u)}$;
  end while
Output: Estimated label matrix $Y^{(u)}$

4. Experiments

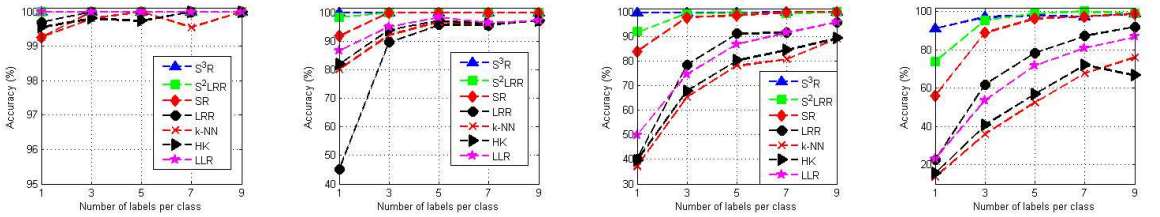
In this section, we evaluate the effectiveness of our proposed STSSL approach on a synthetic data set and several benchmark data sets.

We compare our proposed STSSL with S^{2}LRR and S^{3}R to the single pass SSL with LRR [23], SR [13], and other three popular affinity matrices, including the binary weights (k-NN), the affinity defined by a heat kernel (HK) [6], and the affinity induced by local linear representation (LLR) [25]. For LLR, LRR, and SR, the affinity matrix A is induced from the representation matrix C via using $\frac{1}{2}(|C|+|C^\top|)$. Given the affinity matrix A, the unknown labels are inferred by using the label propagation via the harmonic function approach (LPHF) [34]. For the local methods, including k-NN, the heat kernel, and LLR, we report the best results over $k \in \{5, 10, 15, \cdots, 100\}$ separately for each setting. For heat kernel, we take the distance of each data point to its k-th nearest neighbor as the local bandwidth parameter. For the counterpart algorithms, LRR vs. S^{2}LRR and SR vs. S^{3}R, we keep the ADMM parameters and λ the same, respectively. In our STSSL approach, we set the maximum number of iterations as $T_{max} = 10$ and select $\frac{N}{10}$ seed labels per iteration, where N is the number of samples. For simplicity, we refer to the two instances of our STSSL approach – STSSL+S^{2}LRR and STSSL+S^{3}R – directly as S^{2}LRR and S^{3}R, respectively.

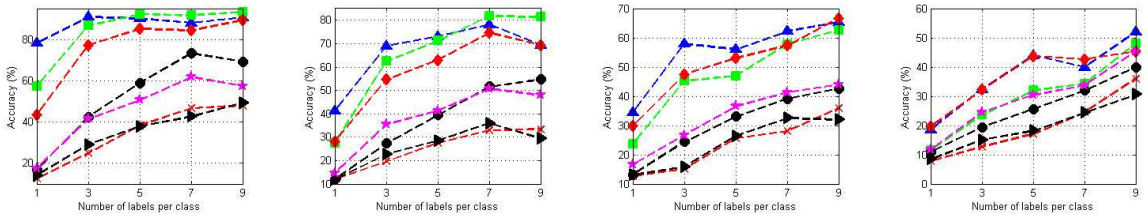
4.1. Experiments on Synthetic Data

Data Preparing. We sample 150 data points from 15 linear subspaces of dimension 5. For subspace S_j, we sample 10 data points by $X_j = U_j Y_j$, where the entries of $Y_j \in \mathbb{R}^{5\times 10}$ are i.i.d. samples from a standard Gaussian and U_j is the left singular matrix computed from a random matrix $R_j \in \mathbb{R}^{100\times 100}$. We then corrupt a certain percentage $p = 10 - 70\%$ of entries uniformly at random.

We repeat each experiment for 10 trials and record the averaged accuracy with different number of labels under different corruptions. Experimental results are presented in Fig. 2. As can be seen from Fig. 2 (a), when the data are corruption-free, all methods perform almost equally good. When the corruptions get heavier as in Fig. 2 (b)-(h), how-



(a) corruptions: 0% (b) corruptions: 10% (c) corruptions: 20% (d) corruptions: 30%



(e) corruptions: 40% (f) corruptions: 50% (g) corruptions: 60% (h) corruptions: 70%

Figure 2. Classification accuracy (%) vs. the number of labels under different percentages of corruptions.

ever, the performance of all methods decrease rapidly, except for S^3R, S^2LRR, and SR. Notice that S^3R and S^2LRR outperform significantly their counterpart algorithms, SR and LRR. These results confirm the robustness and superiority of our proposed STSSL approach. The robustness comes from the explicit modeling of the noise or corruption by the term $\|E\|_\ell$; whereas the superiority attributes to the feedback of weakly supervised information.

To gain some insights to the advantages of our proposed STSSL, we display in Fig. 3 the representation matrices which are computed with SR, LRR, S^2LRR, and S^3R, respectively. The corruption level is 30% and the number of labels is 7. As can be observed, the semi-supervised representations show clearer block diagonal structures.

4.2. Experiments on Benchmark Data Sets

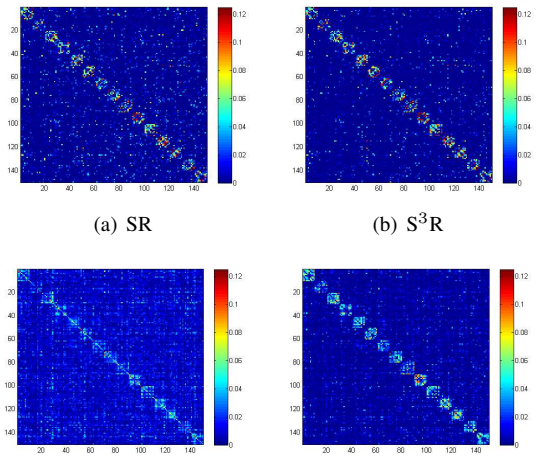
To evaluate the performance of each methods quantitatively, we conduct experiments on the following benchmark data sets.

ORL. The ORL face recognition data set consists of 400 samples from 40 individuals. In our experiments, we take all the 400 samples and resize each image into 28×23.

Yale. The Yale face recognition data set consists of 165 gray-scale images of 15 individuals and there are 11 images per subject. We take all the 165 samples and resize each images into 32×32.

Extended Yale B. The Extended Yale B data set [14] contains 2,414 frontal face images of 38 subjects, with approximately 64 frontal face images per subject. We take all the samples of 38 subjects and resize each image into 32×32.

CMU PIE. The CMU pose, illumination, and expression (PIE) database contains more than 40,000 facial images of 68 people. The data we used consist of 12 subjects and 170 images per subject from varying illuminations and facial expressions, in which each image is resized into 32×32.



(a) SR (b) S^3R

(c) LRR (d) S^2LRR

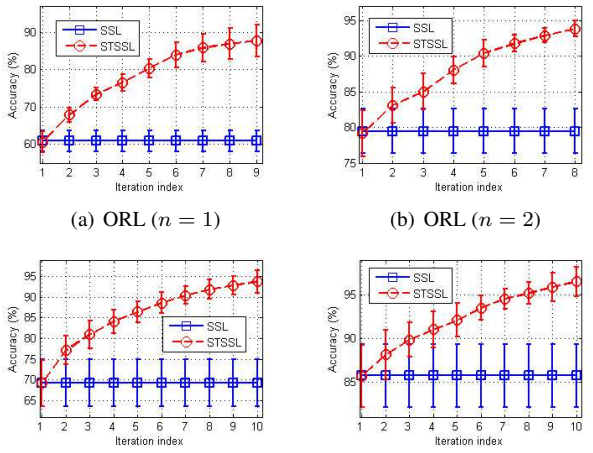
Figure 3. Visualization of different representation matrices.

We compare our proposed STSSL with S^2LRR and S^3R to the single pass SSL with LRR [23], SR [13], and other three popular affinity matrices, including the binary weights (k-NN), the affinity defined by a heat kernel [6], and the affinity induced by LLR [25, 28]. Experimental results are listed in Table 1. We can observe that our proposed STSSL approach with S^2LRR and S^3R outperforms the single pass two-stage SSL with LRR and SR, respectively. These results again confirm the effectiveness of our proposed unified optimization framework.

To gain more insights to the advantages of our proposed STSSL framework, we take SR vs. S^3R as example to show the classification accuracy curves as a function of the num-

Datasets	# Labels	k-NN	Heat Kernel	LLR	SR	LRR	S^{2}LRR	S^{3}R
ORL	1	79.03	69.28	77.92	78.67±1.82	57.25±5.29	85.17±4.25	**87.72**±2.82
	3	81.43	82.68	92.32	91.89±2.05	87.00±2.02	92.82±1.71	**96.64**±1.61
	5	86.35	88.20	96.15	95.75±1.64	92.35±1.31	94.50±1.70	**97.55**±1.19
	7	89.25	91.75	98.50	96.92±0.79	94.58±1.32	96.33±1.12	**98.42**±1.00
Yale	1	36.73	36.47	40.53	39.87±4.19	33.13±4.92	**48.33**±9.28	46.93±6.59
	3	53.50	52.42	59.58	59.25±4.57	60.00±3.02	**72.50**±5.05	68.50±5.07
	5	60.89	59.78	67.00	68.78±4.27	69.33±3.60	**80.33**±3.23	73.11±2.91
	7	63.83	62.67	74.67	76.33±3.83	77.33±5.57	**82.83**±6.09	76.33±4.43
Extended Yale B	5	46.11	46.10	55.45	88.08±0.94	38.91±2.50	74.12±4.86	**96.06**±0.82
	10	54.02	53.89	64.18	92.59±0.84	60.79±1.06	82.10±2.83	**97.64**±0.16
	15	57.96	58.29	72.28	94.62±0.84	72.85±1.18	85.56±1.58	**97.34**±0.32
	20	61.90	62.19	81.23	96.07±0.73	82.06±1.14	87.45±1.12	**97.46**±0.28
CMU PIE	5	54.04	54.41	56.54	69.20 ±5.63	19.46±2.46	55.25±4.09	**93.72**±2.71
	10	65.80	66.55	68.66	85.72 ±3.58	33.37±3.33	68.82±3.42	**96.50**±1.72
	15	73.94	74.28	76.38	92.68 ±1.25	42.58±3.29	80.51±4.47	**97.05**±0.35
	20	78.59	79.00	80.82	94.44 ±1.24	52.49±1.52	88.12±2.14	**97.27**±0.69

Table 1. Classification Accuracy (%) on Benchmark Data Sets. The best results are in bold font.



(a) ORL ($n = 1$) (b) ORL ($n = 2$)

(c) PIE ($n = 5$) (d) PIE ($n = 10$)

Figure 4. Average accuracy (%) with standard variance vs. the iteration number under different number of given labels.

ber of iterations on data sets ORL and PIE with different number of given labels. Each experiment is repeated for 10 trials and the average accuracy and standard variance are recorded. Experimental results are displayed in Fig. 4. As can be seen, the classification accuracy is significantly boosted during the iterations. The performance gains in STSSL derive from three aspects: a) the given labels are used for learning the affinity, b) the inferred labels with the given labels are combined to refine the affinity from the second iteration, and c) the inferred labels are used to seed a better initialization for label propagation. As mentioned in Section 2.1, if the constructed affinity matrix is exactly block-diagonal, with each block being connected and corresponding to a single class, then even a single label per class is able to yield correct classification. In practice, however, the affinity matrix is imperfect and thus using the given labels and the inferred labels to refine the affinity towards the correct block-diagonal would help the label propagation. Moreover, the given labels and a proportion of the inferred labels are combined together to seed a better initialization for label propagation. Putting all together the performance could be improved significantly and the sensitivity of label propagation could be alleviated. As can be observed in Fig.4 (except for (a), in which only a single label is given) that, the iterations tend to lower the variances on each curve, which confirms the latter point.

5. Conclusion

We formulated the existing two-stage SSL problem into a unified optimization framework – termed as Self-Taught Semi-Supervised Learning (STSSL), in which both the given labels and the estimated labels are incorporated to refine the affinity matrix and to facilitate the unknown label estimation. We solved the unified optimization problem efficiently via a combination of an alternating direction method of multipliers with label propagation. Experiments on a synthetic data set and several benchmark data sets demonstrated that the classification accuracy could be significantly boosted by proper feedback of the weakly supervised information during the iterations.

As the future work, various strategies used in active learning and self-learning [24, 2] are worth to be investigated in our STSSL framework and the theoretical guarantee is also worth to explore.

Acknowledgment

C.-G. Li, H. Zhang, and J. Guo are supported by the National Natural Science Foundation of China (NSFC) under grant nos. 61175011, 61273217, and 61511130081, and the 111 project under grant no. B08004. Z. Lin is supported by National Basic Research Program of China under grant no. 2015CB352502, NSFC under grant nos. 61272341 and 61231002, and Microsoft Research Asia Collaborative Research Program. Z. Lin is the corresponding author.

References

[1] Y. Altun, D. A. McAllester, and M. Belkin. Margin semi-supervised learning for structured variables. In *NIPS*, 2005. 1, 2

[2] O. M. Aodha, N. D. Campbell, J. Kautz, and G. J. Brostow. Hierarchical subquery evaluation for active learning on a graph. In *CVPR*. 2014. 8

[3] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm with application to wavelet-based image deblurring. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, number 693-696, 2009. 5

[4] M. Belkin, I. Matveeva, and P. Niyogi. Regularization and semi-supervised learning on large graphs. In *COLT*, 2004. 1

[5] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003. 1

[6] M. Belkin and P. Niyogi. Semi-supervised learning on riemannian manifolds. *Machine Learning*, 56(1-3):209–239, 2004. 1, 2, 6, 7

[7] M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7:2399–2434, 2006. 1

[8] Y. Bengio, O. Delalleau, and N. Le Roux. Label propagation and quadratic criterion. In O. Chapelle, B. Schölkopf, and A. Zien, editors, *Semi-Supervised Learning*, pages 193–216. MIT Press, 2006. 1, 3

[9] J.-F. Cai, E. J. Candés, and Z. Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal of Optimization*, 20(4):1956–1982, 2008. 5

[10] B. Cheng, J. Yang, S. Yan, Y. Fu, and T. Huang. Learning with ℓ_1-graph for image analysis. *IEEE Trans. on Image Processing*, 19(4), 2010. 1, 2

[11] H. Cheng, Z. Liu, and J. Yang. Sparsity induced similarity measure for label propagation. In *ICCV*, 2009. 1, 2, 4

[12] C. Cortes and M. Mohri. On transductive regression. In *NIPS*, 2007. 1

[13] E. Elhamifar and R. Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11):2765–2781, 2013. 4, 6, 7

[14] A. Georghiades, P. Belhumeur, and D. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):643–660, 2001. 7

[15] R. He, W.-S. Zheng, B.-G. Hu, and X.-W. Kong. Nonnegative sparse coding for discriminative semi-supervised learning. In *CVPR*, 2011. 1, 2, 4

[16] M. Herbster, M. Pontil, and L. Wainer. Online learning over graphs. In *ICML*, 2005. 1, 2

[17] T. Joachims. Transductive learning via spectral graph partitioning. In *ICML*, 2003. 1, 2

[18] F. Kang, R. Jin, and R. Sukthankar. Correlated label propagation with application to multi-label learning. In *CVPR*, 2006. 1

[19] T. Kato, H. Kashima, and M. Sugiyama. Robust label propagation on multiple networks. *IEEE Trans. Neural Networks*, 20(1):35–44, 2009. 1, 2

[20] C.-G. Li, J. Guo, and H.-G. Zhang. Learning bundle manifold by double neighborhood graphs. In *ACCV*, 2009. 4

[21] Z. Lin, R. Liu, and Z. Su. Linearized alternating direction method with adaptive penalty for low rank representation. In *NIPS*, 2011. 5

[22] G. Liu, Z. Lin, S. Yan, J. Sun, and Y. Ma. Robust recovery of subspace structures by low-rank representation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 35(1):171–184, Jan 2013. 4

[23] G. Liu, Z. Lin, and Y. Yu. Robust subspace segmentation by low-rank representation. In *ICML*, 2010. 1, 2, 4, 6, 7

[24] M. Loog. Semi-supervised linear discriminant analysis through moment-constraint parameter estimation. *Pattern Recognition Letter*, 27:24–31, 2014. 8

[25] L. K. Saul and S. T. Roweis. Think globally, fit locally: unsupervised learning of low dimensional manifolds. *Journal of Machine Learning Research*, 4:119–155, 2003. 1, 2, 4, 6, 7

[26] M. Szummer and T. Jaakkola. Partially labeled classification with markov random walks. In *NIPS*, 2001. 1, 2

[27] F. Wang and C. Zhang. Label propagation through linear neighborhoods. In *ICML*, 2006. 4

[28] F. Wang and C. Zhang. Label propagation through linear neighborhoods. *IEEE Transactions on Knowledge and Data Engineering*, 20(1):55–67, 2008. 1, 2, 4, 7

[29] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T. S. Huang, and S. Yan. Sparse representation for computer vision and pattern recognitiong. *Proceedings of the IEEE*, 98(6):1031–1044, 2010. 2

[30] S. Yan and H. Wang. Semi-supervised learning by sparse representation. In *SIAM International Conference on Data Mining*, 2009. 1, 2, 4

[31] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In *NIPS*, 2004. 1, 2, 3, 4

[32] D. Zhou, B. Schölkopf, and T. Hofmann. Semisupervised learning on directed graphs. In *NIPS*, 2005. 1, 2

[33] X. Zhu. Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2008. 1, 2

[34] X. Zhu, Z. Ghahramani, and J. D. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, 2003. 1, 2, 3, 4, 6

[35] X. Zhu, J. Kandola, Z. Ghahramani, and J. Lafferty. Nonparametric transforms of graph kernels for semi-supervised learning. In *NIPS*. 2005. 2

[36] L. Zhuang, H. Gao, Z. Lin, Y. Ma, X. Zhang, and N. Yu. Non-negative low rank and sparse graph for semi-supervised learning. In *CVPR*, 2012. 1, 2, 4