ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-507

Semi-Supervised Group Sparse Representation: Model, Algorithm and Applications

Longwen Gao[1] and **Yeqing Li**[2] and **Junzhou Huang**[3] and **Shuigeng Zhou**[4]

Abstract. Group sparse representation (GSR) exploits group structure in data and works well on many problems. However, the group structure must be manually given in advance. In many practical scenarios such as classification, samples are grouped according to their labels. Constructing a consistent group structure in such cases is not easy. The reasons are: 1) samples may be incorrectly labeled; and 2) label assigning in big data is time-consuming and expensive. In this paper, we propose and formulate a new problem, *semi-supervised group sparse representation* (SS-GSR) to support group sparse representation among both labeled and unlabeled data, while learning a more robust group structure, which can be further exploited to more effectively represent other unlabeled data. We develop a model to tackle the SS-GSR problem, based on the manifold assumption in subspace segmentation that samples in the same group lie close in feature space and span the same subspace. We also propose an alternating algorithm to solve the model. Finally, we validate the model via extensive experiments.

1 INTRODUCTION

Sparse representation (SR) [18] and group sparse representation (GSR) [24] have been successfully applied to many regression problems [24] and machine learning tasks, such as the classification tasks of images [15, 22], texts [20, 9] and biological data [13, 23]. GSR considers the group structure of data as prior knowledge and benefits from it when the data is consistent with such structure. For example, in most classification tasks, samples can be seen naturally with a group structure, because samples in the same class tend to be grouped together. For such cases, GSR usually outperforms SR [15] because group sparsity works better when the underlying samples are strongly group-sparse [10]. However, GSR requires that the group structure is explicitly given in advance, which is implied in the class relationship of labeled samples. In real applications, accurate label information may not be easy to acquire. On the one hand, the samples may be incorrectly labeled. On the other hand, it requires a lot human effort to assign the labels, which is prohibitively expensive for big data. Consequently, a large fraction of data in reality are unlabeled although we know that they should have certain labels. In parallel to semi-supervised learning, if we can exploit the large amounts of unlabeled data in the GSR process, better data representation should be achieved.

With this in mind, in this paper we propose a new problem, *semi-supervised group sparse representation* (SS-GSR) to conduct GSR over a dataset consisting of both labeled and unlabeled samples. The aim is two-fold: 1) representing each sample with respect to the other samples while the representation coefficients are consistent with the underlying group structure of the whole dataset; 2) learning a more robust group structure underlying the dataset via exploiting also the unlabeled samples. SS-GSR is not only a nontrivial advancement but also a significant complement to the traditional GSR that represents unlabeled samples with a dictionary of labeled samples by imposing the group sparsity constraint. SS-GSR performs GSR among labeled and unlabeled data, meanwhile refines the group structure explicitly given in the labeled data by additionally utilizing unlabeled data.

To reveal the underlying group structure of the dataset, we believe that the coefficient matrix should be in a specific form. Manifold assumption and block-diagonal constraint are introduced in subspace segmentation [21] to cluster samples into groups. Samples (labeled and unlabeled) are assumed to be grouped according to their underlying subspace and the distance in the feature space. This assumption allows the block-diagonal constraint on the affinity matrix to find clustering structure among samples [8]. In this paper, we employ the same assumption to the SS-GSR problem and formulate our model with block-diagonal constraint, thus the underlying group structure can be discovered with the block structure in the coefficient matrix. Furthermore, to exploit the group structure of unlabeled data in sparse representation, we construct the affinity matrix using the coefficient matrix and try to maintain the local consistency of group structure among samples according to the affinity matrix as in [27].

Contributions of this paper are as follows:

- We propose the problem of SS-GSR to extend GSR so that unlabeled data can be also exploited in the representation process.
- We formulate our model to automatically learn the underlying group structure by utilizing the manifold structure of data, and develop an efficient algorithm to solve the model.
- We validate our model by extensive experiments of two typical applications. Experimental results show that our model outperforms the existing GSR model and three semi-supervised learning methods (including one proposed recently).

The rest of this paper is organized as follows: Section 2 reviews the traditional group sparse representation (GSR) model, which is the starting point and the most related work to our model proposed in this paper. Section 3.2 presents the new model that is called semi-supervised group sparse representation (SS-GSR). Section 4 intro-

[1] Shanghai Key Lab of Intelligent Information Processing, and School of Computer Science, Fudan University, Shanghai 200433, China. Email: lwgao@fudan.edu.cn
[2] University of Texas at Arlington, Arlington, Texas, USA, 76019. Email: yeqing.li@mavs.uta.edu
[3] University of Texas at Arlington, Arlington, Texas, USA, 76019. Email: jzhuang@uta.edu
[4] Correspondence author. Shanghai Key Lab of Intelligent Information Processing, and School of Computer Science, Fudan University, Shanghai 200433, China. Email: sgzhou@fudan.edu.cn

duces the algorithm to solve the proposed model. Section 5 gives the validation of the proposed model on two applications. Section 6 concludes this paper.

2 GROUP SPARSE REPRESENTATION (GSR)

GSR explores group structure information during representation by requiring the coefficients corresponding to different groups to be sparse. The training samples used to represent other samples together constitute a dictionary $\boldsymbol{X} \in \mathcal{R}^{d\times n}$. Let $\mathcal{G}_g$ be the group of indices of training samples with group id $g \in \{1..c\}$, given another test sample $y \in \mathcal{R}^d$, GSR can be formulated as:

$$\min_{\boldsymbol{z}\in\mathcal{R}^n} \frac{1}{2}\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{z}\|_2^2 + \lambda\sum_{g=1}^{c}\left\|\boldsymbol{z}_{\mathcal{G}_g}\right\|_2, \tag{1}$$

with tradeoff parameter $\lambda>0$. Here, the non-zero elements of vector $\boldsymbol{z}_{\mathcal{G}_g}$ are the same as those of vector $\boldsymbol{z}$ indexed in $\mathcal{G}_g$. The first term is the regression error, and the second term can be seen as an ℓ_{21}-norm: the ℓ_2-norm is for the elements of the coefficient vector $\boldsymbol{z}$ inside each group and ℓ_1-norm measures the sparsity among groups.

Given the group structure, GSR favors the selection of multiple correlated samples in the dictionary to represent the test sample, and thus promotes the representation of the test sample in terms of all the training samples from the correct group [15]. Even though, it requires that the group structure of the whole dictionary should be given in advance and the structure should be correct according to the prior knowledge. However, in practice the given structure might not be fully consistent with data due to the complexity of data and noise in data collection. Assigning structures to all samples in the dictionary via human labor might be prohibitively expensive if not impossible when large amount of data are collected. In this paper, we will try to exploit the group structure information of some samples properly and learn the group structure of all samples automatically.

3 SS-GSR MODEL

3.1 Problem statement

Suppose we have a dataset $\boldsymbol{X} \in \mathcal{R}^{d\times n}$ whose column vector $\boldsymbol{X}^i$ $(i = 1, \ldots, n)$ corresponds to each of the n samples. These samples can be grouped into c non-overlapping groups. However only part of these samples are given with group labels in $\mathcal{C} = \{1, \ldots, c\}$, and for the rest of them, the group labels are unknown. We simply assume that the first m samples $\boldsymbol{X}^{1\ldots m}$ are given with group labels, and these group labels form a group label vector $\boldsymbol{G} \in \mathcal{C}^m$. Our problem is to decide a coefficient matrix $\boldsymbol{Z} \in \mathcal{R}^{n\times n}$, whose columns are representation coefficients $\boldsymbol{Z}^i$ that represent sample $\boldsymbol{X}^i$ using the others, and the non-zero elements in $\boldsymbol{Z}^i$ should correspond to samples in the same underlying group with sample $\boldsymbol{X}^i$. That is, the group sparsity on $\boldsymbol{Z}^i$ should be the underlying group sparsity. Since we do not want samples to represent themselves, we fix the diagonal elements in $\boldsymbol{Z}$ to be 0 to avoid such trivial representations. Accordingly, we have the following equation:

$$\boldsymbol{X} = \boldsymbol{X}\boldsymbol{Z},\ s.t.\ \boldsymbol{Z}_i^i = 0,\ \forall i \in \{1, \ldots, n\}. \tag{2}$$

If we rearrange the samples to an order that the samples in the same underlying group are put together, the desired coefficient matrix $\boldsymbol{Z}$ would be a block-diagonal matrix with each block corresponding to a group structure. This gives us the inspiration that we might be possible to find the underlying group structure by finding the block structure in the coefficient matrix $\boldsymbol{Z}$.

However, when we assume the given group structure of $\boldsymbol{X}^{1\ldots m}$ to be unreliable, we need some other assumptions on data that can help find the block structure in $\boldsymbol{Z}$. Interestingly, the works of subspace segmentation [21, 7, 14, 8] follow similar idea to build a block-diagonal affinity matrix $\boldsymbol{W} \in \mathcal{R}_+^{n\times n}$. Subspace segmentation is to segment the samples according to the manifold assumption. The work of [8] explicitly imposes a fixed rank constraint on the graph Laplacian, which constrains the number of connected components in the affinity matrix $\boldsymbol{W}$ as:

$$rank\left(\boldsymbol{L}_{\boldsymbol{W}}\right) = n - c, \tag{3}$$

where $\boldsymbol{L}_{\boldsymbol{W}}$ is the Laplacian matrix for $\boldsymbol{W}$ and c is the number of connected components (a connected component corresponds to a group of samples). Thus the optimal affinity matrix is constrained to be a c-block-diagonal matrix.

Here, we employ the manifold assumption and the block constraint into GSR, with which the underlying group structure can be obtained by finding the block structure of the coefficient matrix. Note that though we used a similar block constraint form to that in [8], our work is different from subspace segmentation [8] at least in two aspects: a) The task is different. Their work aims at solving an unsupervised learning problem, while ours aims at extending the traditional group sparse representation, and applied it to both supervised and semi-supervised learning problems. b) The solution is different. Their work follows a two-step scheme: first they compute an affinity matrix $\boldsymbol{W}$ from data, and then perform regular clustering on the affinity matrix. And the first step is independent from the second step. However in our work, the second step (classification) also affects the first step (computing the affinity matrix). Thus, we propose a combined scheme that solves the affinity matrix and the label assignment jointly, in order to simultaneously obtain a better affinity matrix and a more accurate label assignment.

3.2 Model formulation

We introduce a confidence matrix $\boldsymbol{F} \in \mathcal{R}_+^{n\times c}$ whose elements indicate the probability that a sample belongs to a certain group. So we have the following equation and inequation:

$$\sum_{j=1}^{c} F_i^j = 1, \forall i \in \{1, \ldots, n\},$$
$$0 \le F_i^j \le 1, \forall i \in \{1, \ldots, n\}, j \in \{1, \ldots, c\}.$$

As the first m samples' labels are already known, thus:

$$\boldsymbol{F}_i^{\boldsymbol{G}_i} = 1, \forall i \in \{1, \ldots, n\};$$
$$\boldsymbol{F}_i^j = 0, \forall j \neq \boldsymbol{G}_i, j \in \{1, \ldots, c\}, i \in \{1, \ldots, n\}.$$

The first m rows of $\boldsymbol{F}$ are fixed and we write them as $\boldsymbol{F}_L$. The rest part of $\boldsymbol{F}$ is denoted by $\boldsymbol{F}_U$. Thus $\boldsymbol{F} = [\boldsymbol{F}_L^\top \boldsymbol{F}_U^\top]^\top$.

We take two steps to formulate our model according to the two problems in GSR: first we try to detect the underlying group structure in samples whose group structure is given, then we present the procedure of finding the hidden group structure of the whole set of samples by taking also the unlabeled samples into consideration.

3.2.1 Detecting the underlying structure of labeled data

First, we focus on detecting the underlying structure in labeled data. To avoid the influence of unlabeled data, we fix the coefficients corresponding to those samples as 0, namely $\boldsymbol{Z}_{m+1,\ldots,n}^{m+1,\ldots,n} = \boldsymbol{0}$, and denote

the Laplacian from labeled data to labeled data as $\widehat{\boldsymbol{L}}_{\boldsymbol{W}} = \boldsymbol{L}_{\boldsymbol{W}_{1,\ldots,m}^{1,\ldots,m}}$, where $\boldsymbol{L}_{\boldsymbol{W}}$ is the Laplacian matrix of the affinity matrix $\boldsymbol{W}$.

Based on the requirement in Equation (2) and our assumption in Section 3.1, as well as the group sparsity regularization term, we formulate our model as:

$$\min_{\boldsymbol{Z},\boldsymbol{F}} \frac{1}{2}\|\boldsymbol{X}-\boldsymbol{X}\boldsymbol{Z}\|_F^2 + \lambda\sum_{i=1}^{m}\left\|\boldsymbol{Z}^i\right\|_{G(\boldsymbol{F})} + \gamma tr(\boldsymbol{F}_L^\top \widehat{\boldsymbol{L}}_{\boldsymbol{W}}\boldsymbol{F}_L)$$
$$s.t.\ rank(\widehat{\boldsymbol{L}}_{\boldsymbol{W}}) = m-c, \boldsymbol{W} = (|\boldsymbol{Z}|+|\boldsymbol{Z}|^\top)/2. \quad (4)$$

Above, the norm $\|\cdot\|_{G(\boldsymbol{F})}$ is a group sparse norm in which the group structure is given by the column number of the maximum element of each row in $\boldsymbol{F}$, namely, samples with the same column number are considered to be in the same group.

The first term in the objective function is to ensure the representation to be of small residual error to meet the requirement in Equation (2). The second term uses the confidence matrix to help decide the group structure of the coefficient matrix, and the third term uses the Laplacian matrix constructed by the coefficient matrix as the affinity matrix and perform label propagation on Laplacian graph. The rank constraint is equivalent to a block-diagonal constraint on $\boldsymbol{W}$ [8], which encourages the samples to be clustered into groups according to the manifold assumption.

3.2.2 *Finding the group structure for the entire dataset*

Next, we go to find the hidden group structure of the whole set of samples and use the learned structure information for group sparse representation. Since we only have the group structure of part of the samples, we have to propagate the group structure with respect to the Laplacian $\boldsymbol{L}_{\boldsymbol{W}}$ as in the works of graph-based semi-supervised learning (SSL) [4, 27, 26]. Since in our first step, we have already learned a coefficient matrix and then an affinity matrix can be constructed as $\boldsymbol{W} = (|\boldsymbol{Z}|+|\boldsymbol{Z}|^\top)/2$, the same propagation process as in graph-based SSL can be directly applied to our first model (4). Therefore, we have:

$$\min_{\boldsymbol{Z},\boldsymbol{F}} \frac{1}{2}\|\boldsymbol{X}-\boldsymbol{X}\boldsymbol{Z}\|_F^2 + \lambda\sum_{i=1}^{n}\left\|\boldsymbol{Z}^i\right\|_{G(\boldsymbol{F})} + \gamma tr(\boldsymbol{F}^\top \boldsymbol{L}_{\boldsymbol{W}}\boldsymbol{F})$$
$$s.t.\ rank(\boldsymbol{L}_{\boldsymbol{W}}) = n-c, \boldsymbol{W} = (|\boldsymbol{Z}|+|\boldsymbol{Z}|^\top)/2. \quad (5)$$

In the above formulation, the coefficient matrix $\boldsymbol{Z}$ is learned for all samples, and the rank constraint is performed on the whole Laplacian graph.

Although we have formulated the model, solving the optimization problem is not easy because it is a non-smooth and non-convex problem, and the rank constraint is generally NP-hard. In Section 4, we will present an algorithm to efficiently solve the problem.

3.3 The Advantages of SS-GSR

In our model formulation (5), the affinity matrix $\boldsymbol{W}$ can be divided into four parts:

$$\boldsymbol{W} = \begin{bmatrix} \boldsymbol{W}_{LL} & \boldsymbol{W}_{LU} \\ \boldsymbol{W}_{UL} & \boldsymbol{W}_{UU} \end{bmatrix}, \quad (6)$$

where $\boldsymbol{W}_{UL} = \boldsymbol{W}_{LU}^\top$ indicates the relationship between labeled samples and unlabeled samples, and $\boldsymbol{W}_{LL}$ and $\boldsymbol{W}_{UU}$ indicate respectively the relationships among labeled samples and unlabeled samples. In the best case, all the four matrices are block-diagonal matrices as shown in Figure 1.

Algorithm 1 Iteration between $\boldsymbol{Z}$ and $\boldsymbol{F}$

Input: dictionary $\boldsymbol{X}$, initial labels $\boldsymbol{F}_L$
Initialize $\boldsymbol{W}$
repeat
  Solve $\boldsymbol{F}_U$ using Equation (9)
  Solve $\boldsymbol{Z}$ using Algorithm 2
  Update $\boldsymbol{W} = (|\boldsymbol{Z}|+|\boldsymbol{Z}|^\top)/2$
until $\boldsymbol{Z}$ and $\boldsymbol{F}$ converge

Algorithm 2 Projected subgradient descent

Input: dictionary $\boldsymbol{X}$, labels $\boldsymbol{F}$, initial $\boldsymbol{Z}_{init}$
Initialize step size η
repeat
  Calculate subgradient $\boldsymbol{g}$ of the objective in Equation (7)
  Subgradient descent $\boldsymbol{Z} = \boldsymbol{Z} - \eta\boldsymbol{g}$
  Project $\boldsymbol{Z} = \Pi_{\mathcal{K}}(\boldsymbol{Z})$ as in Algorithm 3
until $\boldsymbol{Z}$ converges

We discuss the advantages of our model from two aspects: 1) learning the underlying group structure that is consistent with the labeled samples; 2) finding the underlying group structure by exploiting both labeled and unlabeled samples, and using the structure to represent all samples.

For the first aspect, how a sample is consistent with its group structure can be measured by the group sparsity of the corresponding column of $\boldsymbol{W}_{LL}$. Since our model learns the underlying group structure automatically, the properly learned $\boldsymbol{W}_{LL}$ will show us how samples are consistent with their group structures and therefore improves the group sparse representation. We take the supervised classification task as an example, where unlabeled samples are classified one by one. For our model, only one sample is unlabeled (the one to be classified), and the matrix $\boldsymbol{W}_{UU}$ becomes a single real number which is set to 0 since it is also the diagonal element, that is, we solve the model (4). In this case, though $\boldsymbol{W}_{UU}$ will not help in classifying the unlabeled samples, our model can still outperforms GSR via learning $\boldsymbol{W}_{LL}$. Obviously, GSR can be seen as a special case of our model when the matrix $\boldsymbol{W}_{LL}$ is fixed as an zero matrix.

For the second aspect, we compare our model with graph-based *semi-supervised learning* (SSL) methods because they all use the group information of unlabeled samples. The major difference is that, for the graph-based SSL methods, the affinity matrix $\boldsymbol{W}$ must be constructed in advance and is fixed during the learning process. However, in our model the affinity matrix is constructed by coefficient matrix $\boldsymbol{Z}$, which is learned during the model optimization. Furthermore, $\boldsymbol{W}$ in our model contains the underlying structure of data, while a pre-given $\boldsymbol{W}$ in graph-based SSL methods may not be consistent with the structure of data [25]. In Section 5, our experiments over five real datasets clearly show that our model outperforms the graph-based SSL methods even when similar initialized $\boldsymbol{W}$ is used.

4 SS-GSR ALGORITHM

In this section, we first design an alternating algorithm to solve the proposed model, and then briefly discuss the convergence of the algorithm.

4.1 Alternatively solving Z and F

Note that the rank constraint is all about the coefficient matrix $\boldsymbol{Z}$, so we first alternate between solving $\boldsymbol{Z}$ and solving $\boldsymbol{F}$ as outlined in

Algorithm 3 Projecting $\boldsymbol{Z}$ into $\mathcal{K}$

Input: $\boldsymbol{Z}_0$ and c
Initialize $\boldsymbol{Z} = \boldsymbol{Z}_0$; $\rho = 1.1$; $\beta = 1 \times 10^{-4}$
repeat
  Solve the first quadratic problem in (13) for $\boldsymbol{Z}$
  Calculate $\widetilde{\boldsymbol{L}}$ via Equation (14)
  Update $\boldsymbol{J} = \boldsymbol{J} + \beta(\widetilde{\boldsymbol{L}} - \boldsymbol{L}_{\boldsymbol{W}})$
  Update $\beta = \rho\beta$
until $\boldsymbol{Z}$ and $\widetilde{\boldsymbol{L}}$ converge

Algorithm 1.

When $\boldsymbol{F}$ is fixed, the optimization problem becomes:

$$\min_{\boldsymbol{Z}} \frac{1}{2} \|\boldsymbol{X} - \boldsymbol{X}\boldsymbol{Z}\|_F^2 + \lambda \sum_{i=1}^{n} \left\|\boldsymbol{Z}^i\right\|_{G(\boldsymbol{F})} \quad s.t.\ rank\left(\boldsymbol{L}_{\boldsymbol{W}}\right) = n - c. \tag{7}$$

We will further show how to solve this in Algorithm 2.

When $\boldsymbol{Z}$ is fixed, the optimization problem becomes:

$$\min_{\boldsymbol{F}} tr\left(\boldsymbol{F}^\top \boldsymbol{L}_{\boldsymbol{W}} \boldsymbol{F}\right), \tag{8}$$

and this unconstrained problem has a closed form solution [27]:

$$\boldsymbol{F}_U = \left(\boldsymbol{D}_{UU} - \boldsymbol{W}_U U\right)^{-1} \boldsymbol{W}_{UL} \boldsymbol{F}_L, \tag{9}$$

where $\boldsymbol{D}$ is a diagonal matrix whose diagonal elements are the sums of every row of $\boldsymbol{W}$ and $\boldsymbol{D}_{UU}$ is that of $\boldsymbol{W}_{UU}$.

4.2 Sub-gradient descent for $\boldsymbol{Z}$

The optimization problem (7) is a non-smooth and constrained problem, which can be solved by the projected subgradient descent method. Let $\mathcal{K}$ be the set of c-block-diagonal matrix as:

$$\mathcal{K} = \{\boldsymbol{Z} | rank(\boldsymbol{L}_{\boldsymbol{W}}) = n - c\}. \tag{10}$$

Thus the rank constraint can be rewritten as $\boldsymbol{Z} \in \mathcal{K}$. For each iteration, we perform a subgradient descent on $\boldsymbol{Z}$ and then project $\boldsymbol{Z}$ back into the feasible set $\mathcal{K}$. This process is shown in Algorithm 2.

4.3 Projecting $\boldsymbol{Z}$ into $\mathcal{K}$

To distinguish between input variable and output variable, we assume that the variable to be projected is $\boldsymbol{Z}_0$. The projection step is to find a matrix in the set $\mathcal{K}$, which is closest to $\boldsymbol{Z}_0$ as follows:

$$\min_{\boldsymbol{Z}} \frac{1}{2} \|\boldsymbol{Z} - \boldsymbol{Z}_0\|_F^2, \ s.t.\ \boldsymbol{Z} \in \mathcal{K}. \tag{11}$$

We introduce an auxiliary variable $\widetilde{\boldsymbol{L}}$ to replace the Laplacian matrix $\boldsymbol{L}_{\boldsymbol{W}}$ and rewrite the projection (11) via Augmented Lagrangian Multiplier as in [1]:

$$\min_{\boldsymbol{Z}, \widetilde{\boldsymbol{L}}} \frac{1}{2} \|\boldsymbol{Z} - \boldsymbol{Z}_0\|_F^2 + \left\langle \boldsymbol{J}, \widetilde{\boldsymbol{L}} - \boldsymbol{L}_{\boldsymbol{W}} \right\rangle + \frac{\beta}{2} \left\|\widetilde{\boldsymbol{L}} - \boldsymbol{L}_{\boldsymbol{W}}\right\|_F^2, \quad s.t.\ rank(\widetilde{\boldsymbol{L}}) = n - c, \tag{12}$$

where $\boldsymbol{J}$ is the Lagrangian multiplier and β is an increasing weight parameter. This problem can be solved by alternatively updating $\boldsymbol{Z}$, $\widetilde{\boldsymbol{L}}$ and $\boldsymbol{J}$ as follows:

$$\begin{aligned} \boldsymbol{Z} &= \operatorname*{argmin}_{\boldsymbol{Z}} \frac{1}{2} \|\boldsymbol{Z} - \boldsymbol{Z}_0\|_F^2 - \langle \boldsymbol{J}, \boldsymbol{L}_{\boldsymbol{W}} \rangle + \frac{\beta}{2} \left\|\widetilde{\boldsymbol{L}} - \boldsymbol{L}_{\boldsymbol{W}}\right\|_F^2; \\ \widetilde{\boldsymbol{L}} &= \operatorname*{argmin}_{\widetilde{\boldsymbol{L}}} \left\langle \boldsymbol{J}, \widetilde{\boldsymbol{L}} \right\rangle + \frac{\beta}{2} \left\|\widetilde{\boldsymbol{L}} - \boldsymbol{L}_{\boldsymbol{W}}\right\| \ s.t.\ rank(\widetilde{\boldsymbol{L}}) = n - c; \\ \boldsymbol{J} &= \boldsymbol{J} + \beta(\widetilde{\boldsymbol{L}} - \boldsymbol{L}_{\boldsymbol{W}}). \end{aligned} \tag{13}$$

The first problem above can be solved via quadratic programming because except for the first term, all the other terms contain only $|\boldsymbol{Z}|$. Thus the sign of all elements in the optimal $\boldsymbol{Z}$ are the same as those of elements in $\boldsymbol{Z}_0$. The second problem has a closed-form solution via SVD [6]:

$$\widetilde{\boldsymbol{L}} = \boldsymbol{U}^{1:(n-c)} \boldsymbol{\Sigma}_{1:(n-c)}^{1:(n-c)} (\boldsymbol{V}^{1:(n-c)})^\top, \tag{14}$$

where $\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top = \boldsymbol{L}_{\boldsymbol{W}} - \frac{1}{\beta}\boldsymbol{J}$. The projection process is outlined in Algorithm 3.

Our algorithm has a relatively higher computational complexity, because we try to solve a much more challenging problem than the existing algorithms. The main challenge is the noise in labeled data and the unknown group structure of a large fraction of unlabeled data, which have not been considered in the existing works. Furthermore, our algorithm is suitable for a branch of accelerating strategies. For example, with stochastic sub-gradient descent, our algorithm can be implemented in a distributed way.

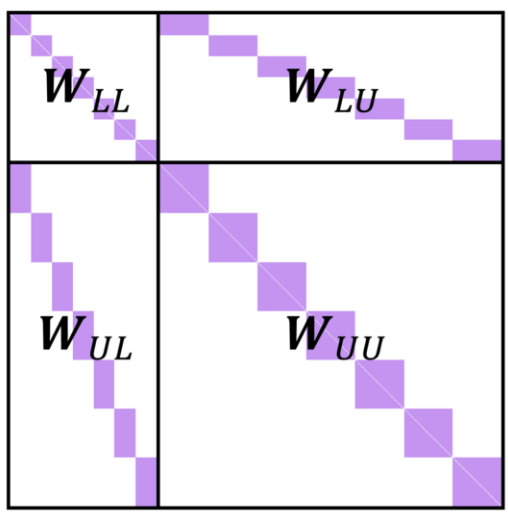

Figure 1. illustration of $\boldsymbol{W}$ learned via SS-GSR.

4.4 Convergence of the algorithm

The optimization problem in Model (5) is strongly non-convex and we solve it using an EM-like algorithm (Algorithm 1). The motivation of using such an algorithm is based on the fact that minimizing the objective with respect to $\boldsymbol{F}$ is obviously a convex problem and minimizing the objective with respect to $\boldsymbol{Z}$ has been approximated using its convex relaxation as in [1]. Therefore, by convex relaxation, the optimization problem actually solved is a bi-convex problem with respect to $\boldsymbol{F}$ and $\boldsymbol{Z}$.

Also, the gradient descent with projection used in solving $\boldsymbol{Z}$ is guaranteed to converge to the global optimum because the Scalable Restricted Isometry Property holds [1, 17].

It is worthy of noting that in our experiments, the algorithm usually gets converged within 10 outer iterations, and achieves satisfactory result.

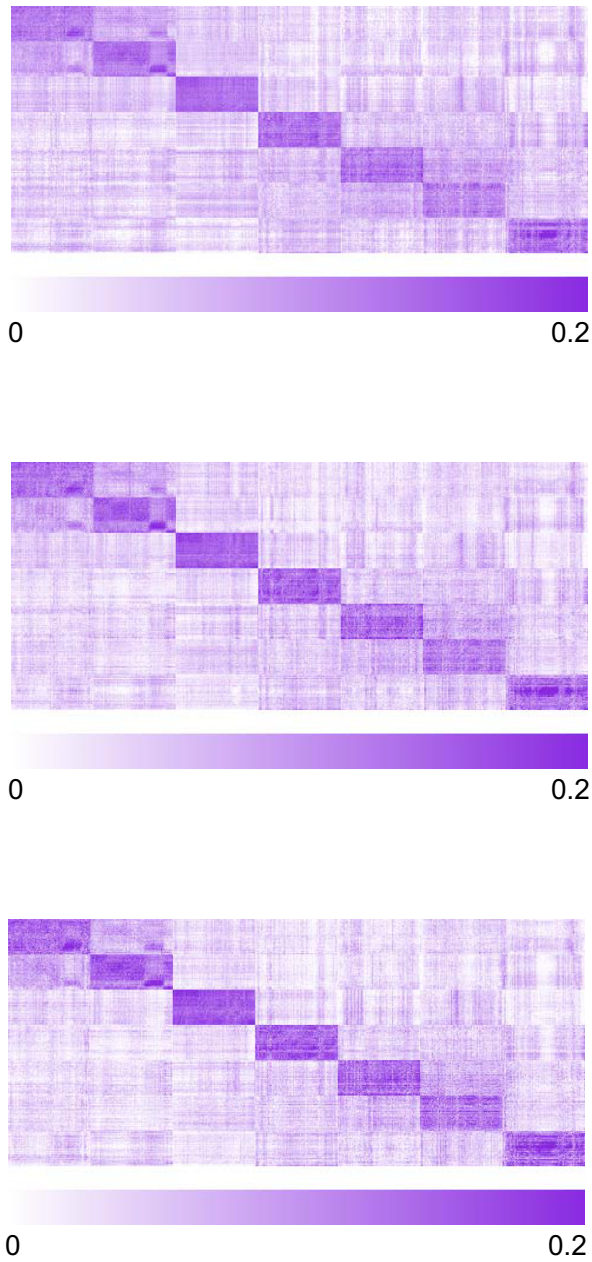

Figure 2. Pictures of Z on Caltech7. From top to bottom are: (a) $\boldsymbol{Z}$ generated by GSR, (b) $\boldsymbol{Z_{LU}}$ generated by SS-GSR-1 and (c) $\boldsymbol{Z}_{LU}$ generated by SS-GSR-2. SS-GSR-1 learns the group structure using only labeled data, while SS-GSR-2 learns the group structure using both labeled and unlabeled data.

Table 1. Group sparsity results of different representation methods on dataset Caltech7. SS-GSR-1 learns the group structure using only labeled data, while SS-GSR-2 learns the group structure using both labeled and unlabeled data.

Representation Method	Group Sparsity
GSR	4455.0
SS-GSR-1	3455.2
SS-GSR-2	**3144.8**

Table 2. Details of datasets used in the experiments.

Dataset	Data type	Num. of samples	Num. of classes
Caltech7	images	1471	7
PENDIGITS	images	5620	10
OPTDIGITS	images	5620	10
Reuters	texts	7424	6
WEBKB4	texts	4196	4

5 Performance Evaluation and Applications

To validate the effectiveness of our method, here we apply it to both supervised and semi-supervised classification tasks. Concretely, we first evaluate the improvement on representing samples with the help of SS-GSR, we then test the performance of SS-GSR on both a supervised classification task and a semi-supervised classification task, and compare it with some major existing methods. We evaluate those methods with performance metrics *Accuracy*, *Precision* and *Recall*, which are first evaluated on each class and then averaged over the classes.

5.1 Model validation: SS-GSR *vs*. GSR

We compare the representation abilities of GSR and SS-GSR on dataset Caltech7 [12] in terms of group sparsity. To calculate the group sparsity, we generate the coefficient matrices that use labeled data to represent the other data with the same parameter. For GSR, it is the whole coefficient matrix $\boldsymbol{Z}$; for SS-GSR, it is the matrix $\boldsymbol{Z}_{LU}$. We generate two results for SS-GSR: SS-GSR-1 represents the test data one by one so that the matrix $\boldsymbol{Z}_{UU}$ is fixed as 0, namely it learns only the group structure of the labeled data as in model (4); SS-GSR-2 represents the test data with both labeled and test data, namely it learns the group structure of all data (labeled and unlabeled) as in model (5).

Figures 2 (a)-(c) show the normalized coefficient matrices. As we have already sorted the samples according to their labels, the expected coefficient matrix should be a block-diagonal matrix. In our above figures, those with fewer non-zero elements outside the diagonal blocks are better representations. We can see that the color of the

Table 3. Supervised classification results. Both algorithms classify test samples one by one.

Dataset	Method	Accuracy	Precision	Recall
Caltech7	GSR	96.33%	87.63%	87.14%
	SS-GSR	**96.73%**	**89.05%**	**88.57%**
PENDIGITS	GSR	99.63%	98.33%	98.13%
	SS-GSR	99.63%	**98.35%**	98.13%
OPTDIGITS	GSR	99.40%	97.32%	97.00%
	SS-GSR	**99.42%**	**97.38%**	**97.10%**
Reuters	GSR	92.67%	65.13%	63.71%
	SS-GSR	**93.71%**	**68.18%**	**66.86%**
WEBKB4	GSR	84.00%	68.00%	68.00%
	SS-GSR	**85.13%**	**71.81%**	**70.25%**

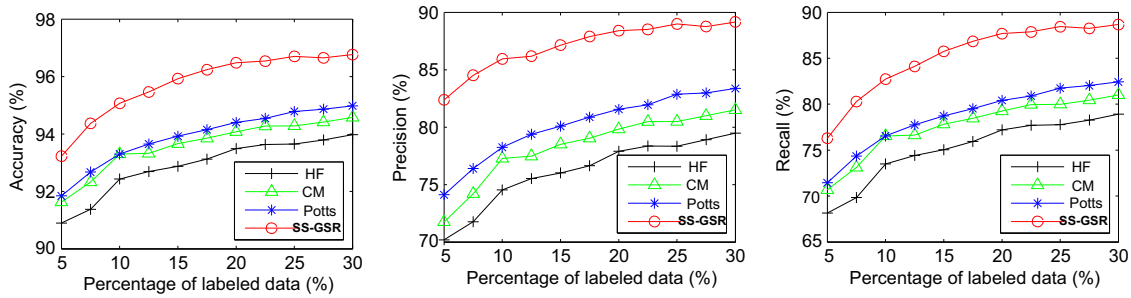

Figure 3. Performance on Caltech7 with 5%-30% labeled samples.

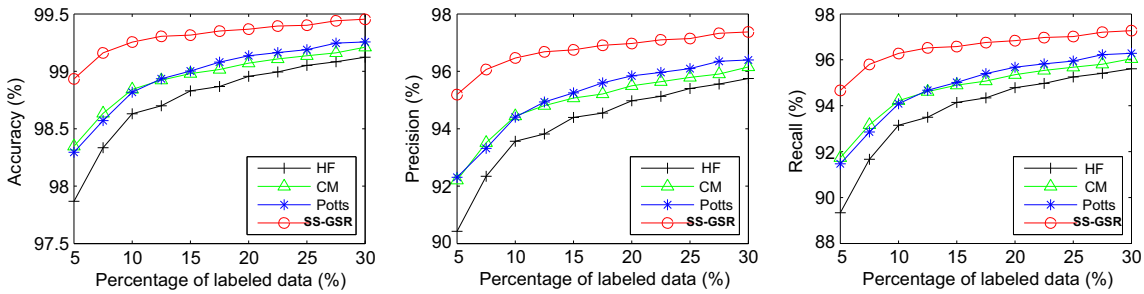

Figure 4. Performance on PENDIGITS with 5%-30% labeled samples.

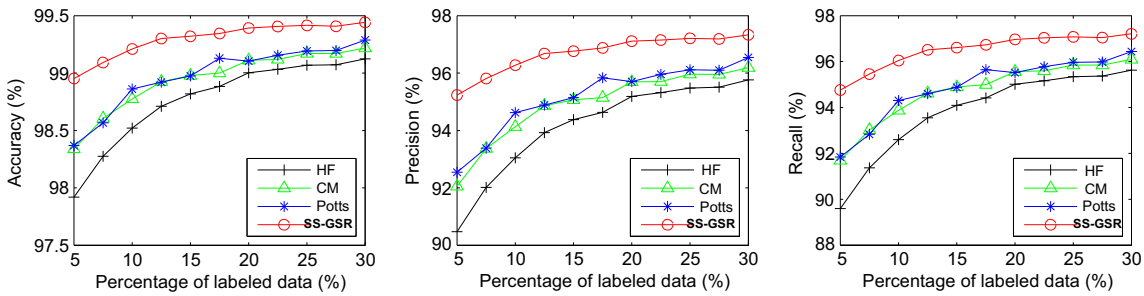

Figure 5. Performance on OPTDIGITS with 5%-30% labeled samples.

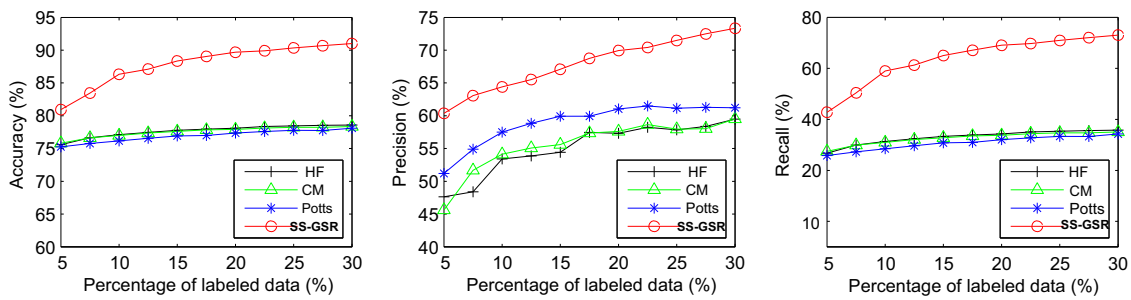

Figure 6. Performance on Reuters with 5%-30% labeled samples.

area outside the diagonal blocks of Figure 2 (a) is obviously deeper than the color of the area outside the diagonal blocks of Figure 2 (b), and the color of the area outside the diagonal blocks of Figure 2 (b) is slightly deeper than the color of the area outside the diagonal blocks of Figure 2 (c). Thus, Figure 2 (c) is better than Figure 2 (b) and Figure 2 (b) is better than Figure 2 (a). We further calculate the group

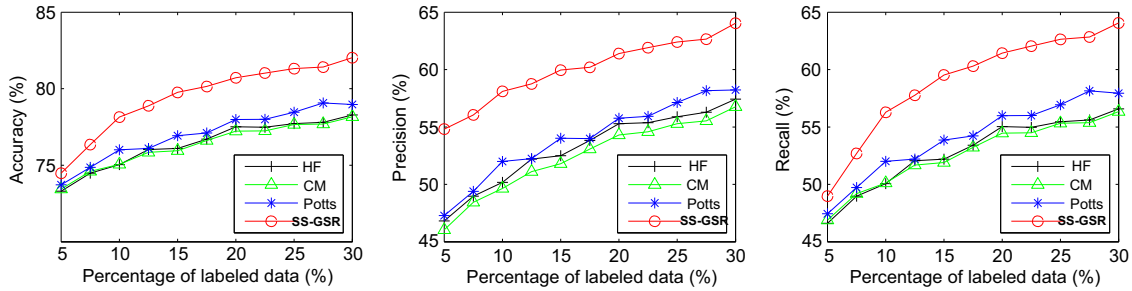

Figure 7. Performance on WEBKB4 with 5%-30% labeled samples.

sparse norm of normalized $\boldsymbol{Z}$ in GSR and normalized $\boldsymbol{Z}_{LU}$ in GSL, and the results are listed in Table 1. By comparing Figure 2 (a), Figure 2 (b) and Figure 2 (c) and checking the results in Table 1, we can conclude that: 1) the coefficient matrices generated by SS-GSR are sparser than the one generated by GSR, which indicates that SS-GSR can more effectively mine and exploit the relationship between the labeled data and the unlabeled data than GSR; 2) As far as sparsity is concerned, learning the structure from the whole data set is better than learning the structure from only the labeled data. The results meet the expectation of our model: exploiting both labeled and unlabeled data in a semi-supervised way can do better group sparse representation.

5.2 Performance comparison in two applications

We apply the new model to two applications: supervised classification and semi-supervised classification, and compare its performance with that of some major existing methods. Five datasets, including Caltech7 [12], PENDIGITS [2], OPTDIGITS [2], Reuters [11] and WEBKB4 [5] are used. The details of these datasets are shown in Table 2. Three performance metrics *Accuracy*, *precision* and *recall* are employed for performance comparison.

5.2.1 Supervised classification task

The first application is text classification, a popular supervised learning task. In this task, our aim is to compare the performance of the traditional GSR model and our new model SS-GSR (when no unlabeled data are used). For each dataset in Table 2, we perform 10-fold cross-validation to compare the classification results of GSR and SS-GSR: give labels to 9 subsets of samples and then use GSR and SS-GSR to classify the rest samples one by one. This process is repeated 10 times and the output results are averaged. The results are shown in Table 3. We can see that SS-GSR works better than GSR in supervised classification. This is because samples in the dictionary are not fully consistent with their labels. For those real datasets, noisy feature vectors can not be given simple labels and outliers cause mislabeling. Nevertheless, SS-GSR can learn a more consistent group structure from the labeled data and selects more precise groups of data according to their underlying group structure.

5.2.2 Semi-supervised classification task

This second application is semi-supervised text classification, a semi-supervised learning task. In this classification task, we compare the capability of our model in group structure learning with that of three typical (including one proposed recently) graph-based semi-supervised methods:

- **Harmonic function (HF)** [27]: it assumes that the harmonic property of label function should be preserved with respect to the graph with given affinity matrix (weight matrix). Equivalently, this method minimizes the quadratic energy function which results in a harmonic solution.
- **Consistency method (CM)** [26]: it proposes a regularization framework which contains two terms: the smoothness term and the fitting term. The former penalizes on the changes between nearby points and the latter penalizes on the change from the given labels. By trading-off between these two terms, the method finds a smooth solution with respect to the intrinsic structure of data points.
- **Mumford-Shah-Potts model (Potts)** [3]: it extends the Mumford-Shah method [16] and Potts method [19] to transductive learning problem using ℓ_1 relaxation.

For each dataset in Table 2, we give labels only to $5\% - 30\%$ random samples (uniformly selected from each class), so the remaining samples are not labeled. We then use the three SSL methods and SS-GSR to decide the labels of the unlabeled samples. This process is repeated 100 times and the output results are averaged. The results of *recall*, *precision* and *accuracy* are presented in Figures 3-7.

From those figures, we can see that SS-GSR clearly outperforms the SSL methods even though they are initialized with the same affinity matrices. There reasons are: on the one hand, manually setting affinity matrix in SSL methods has the consistency problem with real data. On the other hand, the affinity matrix learned by SS-GSR contains the underlying group structure information of all samples and thus is more accurate.

6 CONCLUSION

In this paper, we propose and formulate semi-supervised GSR (SS-GSR) to conduct group sparse representation on datasets containing both labeled and unlabeled data. It can overcome the two drawbacks of the traditional GSR: 1) the pre-defined group structure in GSR may not be fully consistent with that in data; and 2) the underlying group structure of unlabeled data is not exploited in GSR. Compared with GSR, SS-GSR is able to utilize the prior group structure of labeled data properly and take advantage of the group structure information of the unlabeled data. In comparison with SSL methods, SS-GSR can automatically learn the structured affinity matrix from the data instead of using a fixed one. We apply SS-GSR to both supervised and semi-supervised classification tasks, which validate the effectiveness and advantages of SS-GSR.

Acknowledgement

This work was partially supported by the Key Projects of Fundamental Research Program of Shanghai Municipal Commission of Science and Technology under grant No. 14JC1400300.

REFERENCES

[1] Amir Beck and Marc Teboulle, 'A linearly convergent algorithm for solving a class of nonconvex/affine feasibility problems', in *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, 33–48, Springer, (2011).

[2] Catherine Blake and Christopher J Merz, '{UCI} repository of machine learning databases', (1998).

[3] Xavier Bresson, Xue-Cheng Tai, Tony F Chan, and Arthur Szlam, 'Multi-class transductive learning based on ℓ_1 relaxations of cheeger cut and mumford-shah-potts model', *Journal of Mathematical Imaging and Vision*, **49**(1), 191–201, (2014).

[4] Olivier Chapelle, Bernhard Schölkopf, Alexander Zien, et al., *Semi-supervised learning*, volume 2, MIT press Cambridge, 2006.

[5] Mark Craven, Andrew McCallum, Dan PiPasquo, Tom Mitchell, and Dayne Freitag, 'Learning to extract symbolic knowledge from the world wide web', Technical report, DTIC Document, (1998).

[6] Carl Eckart and Gale Young, 'The approximation of one matrix by another of lower rank', *Psychometrika*, **1**(3), 211–218, (1936).

[7] Ehsan Elhamifar and Rene Vidal, 'Sparse subspace clustering: Algorithm, theory, and applications', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **35**(11), 2765–2781, (2013).

[8] Jiashi Feng, Zhouchen Lin, Huan Xu, and Shuicheng Yan, 'Robust subspace segmentation with block-diagonal prior', in *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'14)*, pp. 3818–3825. IEEE, (2014).

[9] Longwen Gao, Shuigeng Zhou, and Jihong Guan, 'Effectively classifying short texts by structured sparse representation with dictionary filtering', *Information Sciences*, **323**, 130–142, (2015).

[10] Junzhou Huang and Tong Zhang, 'The benefit of group sparsity', *The Annals of Statistics*, **38**(4), 1978–2004, (2010).

[11] David D Lewis, 'Reuters-21578 text categorization test collection, distribution 1.0', *http://www. research. att. com/˜ lewis/reuters21578. html*, (1997).

[12] Fei-Fei Li, Rob Fergus, and Pietro Perona, 'Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories', *Computer Vision and Image Understanding*, **106**(1), 59–70, (2007).

[13] Yifeng Li and Alioune Ngom, 'Fast sparse representation approaches for the classification of high-dimensional biological data', in *2012 IEEE International Conference on Bioinformatics and Biomedicine (BIBM'12),*, pp. 1–6. IEEE, (2012).

[14] Guangcan Liu, Zhouchen Lin, Shuicheng Yan, Ju Sun, Yong Yu, and Yi Ma, 'Robust recovery of subspace structures by low-rank representation', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **35**(1), 171–184, (2013).

[15] Angshul Majumdar and Rabab K Ward, 'Classification via group sparsity promoting regularization', in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'09)*, pp. 861 –864, (Apr. 2009).

[16] David Mumford and Jayant Shah, 'Optimal approximations by piecewise smooth functions and associated variational problems', *Communications on pure and applied mathematics*, **42**(5), 577–685, (1989).

[17] Feiping Nie, Xiaoqian Wang, and Heng Huang, 'Clustering and projected clustering with adaptive neighbors', in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 977–986. ACM, (2014).

[18] Bruno A Olshausen and David J Field, 'Sparse coding with an overcomplete basis set: A strategy employed by v1?', *Vision research*, **37**(23), 3311–3325, (1997).

[19] Renfrey Burnard Potts, 'Some generalized order-disorder transformations', **48**(01), 106–109, (1952).

[20] Tara N Sainath, Sameer R Maskey, Bhuvana Ramabhadran, and Dimitri Kanevsky, 'Sparse Representations for Text Categorization', in *INTERSPEECH'10*, pp. 2266–2269, (2010).

[21] René Vidal, 'A tutorial on subspace clustering', *IEEE Signal Processing Magazine*, **28**(2), 52–68, (2010).

[22] John Wright, Allen Y Yang, Arvind Ganesh, Shankar S Sastry, and Yi Ma, 'Robust face recognition via sparse representation', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **31**(2), 210 –227, (Feb. 2009).

[23] Lei Yuan, Alexander Woodard, Shuiwang Ji, Yuan Jiang, Zhi-Hua Zhou, Sudhir Kumar, and Jieping Ye, 'Learning sparse representations for fruit-fly gene expression pattern image annotation and retrieval', *BMC Bioinformatics*, **13**(1), 107, (2012).

[24] Ming Yuan and Yi Lin, 'Model selection and estimation in regression with grouped variables', *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **68**(1), 49–67, (2006).

[25] Richard S Zemel and Miguel Á Carreira-Perpiñán, 'Proximity graphs for clustering and manifold learning', in *Advances in Neural Information Processing Systems*, pp. 225–232, (2004).

[26] Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Schölkopf, 'Learning with local and global consistency', *Advances in Neural Information Processing Systems*, **16**, 321–328, (2004).

[27] Xiaojin Zhu, Zoubin Ghahramani, John Lafferty, et al., 'Semi-supervised learning using gaussian fields and harmonic functions', in *Proceedings of the 20th Annual International Converence on Machine Learning (ICML'03)*, volume 3, pp. 912–919, (2003).