Semi-supervised Learning by Sparse Representation

Shuicheng Yan [*] Huan Wang [†]

Abstract

In this paper, we present a novel semi-supervised learning framework based on ℓ_1 *graph*. The ℓ_1 graph is motivated by that each datum can be reconstructed by the sparse linear superposition of the training data. The sparse reconstruction coefficients, used to deduce the weights of the directed ℓ_1 graph, are derived by solving an ℓ_1 optimization problem on sparse representation. Different from conventional graph construction processes which are generally divided into two independent steps, *i.e.*, adjacency searching and weight selection, the graph adjacency structure as well as the graph weights of the ℓ_1 graph is derived simultaneously and in a parameter-free manner. Illuminated by the validated discriminating power of sparse representation in [16], we propose a semi-supervised learning framework based on ℓ_1 graph to utilize both labeled and unlabeled data for inference on a graph. Extensive experiments on semi-supervised face recognition and image classification demonstrate the superiority of our proposed semi-supervised learning framework based on ℓ_1 graph over the counterparts based on traditional graphs.

1 Introduction

Image classification experienced fast growth over the past decade, and among them classification based on dimensionality reduction techniques shows to be a promising way [3][1][17]. Certain category of images are generally deemed as high dimensional points lying on or nearly on a low dimensional manifolds. Recently many dimensionality reduction algorithms were proposed to preserve local manifold structure and at the same time boost the discriminative power among different classes [17]. Graph is a powerful tool for data analysis, and graph based algorithms are widely used these days in a variety of research areas, such as manifold embedding, semi-supervised learning, and image matching [2][20][14]. Also, graph has proved to be successful in characterizing pairwise data relationship and manifold exploration, and a number of algorithms have been proposed to utilize graph as a tool for designing algorithms for dimensionality reduction, such as ISOMAP [15], LLE [13], and Laplacian Eigenmap [2]. Recently a general framework called graph embedding [17] was proposed and claimed that most traditional dimensionality reduction algorithms can be unified within the general graph embedding framework.

One issue often emerging in real-world applications is the lack of enough labeled training data. Correspondingly, semi-supervised learning was proposed to utilize both labeled data and the information conveyed by the marginal distribution of the unlabeled samples to boost the algorithmic performance [18][8]. Zhu et al. [20] utilized the harmonic property of Gaussian random field over the graph for semi-supervised learning. Belkin and Niyogi [4] learned a regression function that fits the labels at labeled data and at the same time maintains a smoothness over the data manifold expressed by the graph. Zhou et al. [19] proposed to conduct semi-supervised learning with the local and global consistency.

A common assumption for all these graph based algorithms is the well-poseness of the constructed graph. But the algorithmic performance relies heavily on the graph construction process. A number of methods have been proposed for graph construction, among which the popular ones include range graph, k nearest neighbor graph [15], and local linear reconstruction graph proposed in LLE [13]. The traditional graph construction algorithms share the same problem, namely, they all have certain parameters which required manual setting and the parameter has a great impact on the structure of the constructed graph. Therefore, the consequent processes, such as dimensionality reduction, manifold embedding, or semi-supervised learning, are sensitive to these graph parameters and the overall algorithmic robustness is challenged.

Graph is a gathering of pairwise relations while the relation among visual images is essentially an estimation by human cognition system. It has been proved [12] in neural science that the human vision system seeks a sparse coding for the incoming image using a few *words* in a *feature vocabulary*. Olshausen et al. [10] introduced the Bayesian framework to simulate the sparse coding mechanism of human vision system. Wright et al. [16] demonstrated that the ℓ_1 linear reconstruction error minimization can naturally lead to a sparse representation for human facial images. Meinshausen and B$\ddot{u}$hlmannthe's work in [9] is very related with our work in this paper. Their work studies the problem of how to determine the zero entries in the inverse covariance matrix of a multivariate normal distribution corresponds to

[*] Department of Electrical and Computer Engineering, National University of Singapore, Singapore

[†] Computer Science Department, Yale University, USA

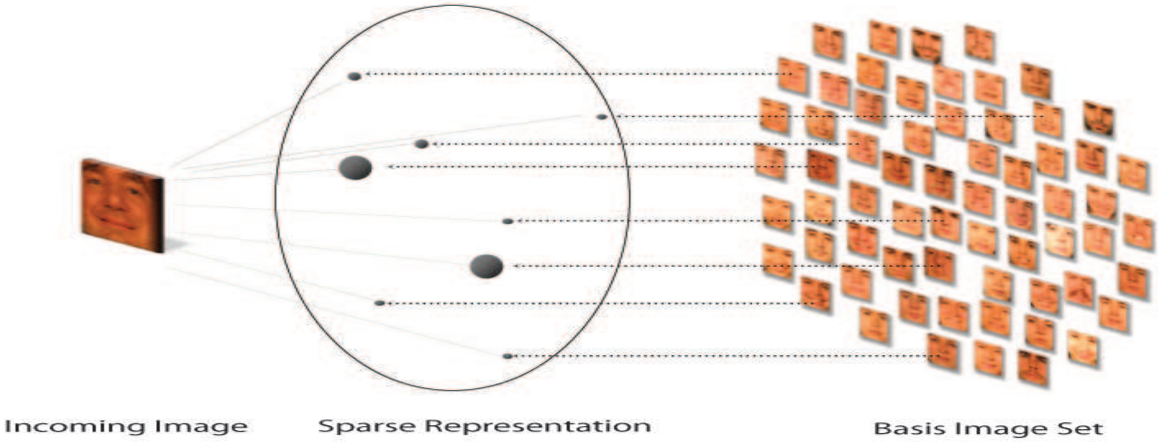


Figure 1: Sparse representation for the incoming image using basis images. The dots demonstrate the coefficients in the linear superposition.

conditional independence restrictions between variables, and shows that neighborhood selection with the Lasso [9] is a computationally attractive alternative to standard covariance selection for sparse high-dimensional graphs. Neighborhood selection estimates the conditional independence restrictions separately for each node in the graph and is hence equivalent to variable selection for Gaussian linear models [9].

In this paper, we present a novel fully automatic algorithm which implements the graph construction in a parameter-free way. The neighborhood relation among images is naturally revealed by an ℓ_1 optimization as shown in Figure 1 and in the meantime, the direction of the edges and the graph weights are also generated during the ℓ_1 optimization process. The sparse coding deduced by ℓ_1 optimization is employed in the graph representation and thus the constructed graph is robust to noise and partial image occlusions. In addition, as an extension of the traditional semi-supervised learning over undirected graph, we present a semi-supervised learning framework on the directed ℓ_1 graph. Extensive experiments on semi-supervised face recognition and image classification show that our proposed ℓ_1 graph can automatically adjust to different databases and semi-supervised learning based on ℓ_1 graph shows to be superior over the counterparts based on traditional graphs.

The remainder of this paper is organized as follows: In Section 2, we give an overview of traditional methods for graph construction, followed by our ℓ_1 graph construction in Section 3. Section 4 presents the semi-supervised learning framework based on directed ℓ_1 graph, and Section 5 demonstrates the detailed experimental results, and we conclude this paper in Section 6.

2 Traditional Graph Construction

For a classification problem, we assume that the training sample data are given as $X = [x_1, x_2, \dots, x_N]$, where $x_i \in \mathbb{R}^d$ and N is the total number of training samples. Traditional graph construction methods typically decompose the graph construction process into two steps, graph adjacency construction and graph weight calculation.

For graph adjacency construction, there exist two widely used methods [2]:

1. ϵ-ball neighborhood. The samples x_i and x_j are considered as neighbors if and only if $\|x_i - x_j\|^2 < \epsilon$, where the norm is the usual Euclidean norm in $\mathbb{R}^d$ and the parameter $\epsilon \in \mathbb{R}$ and $\epsilon > 0$.

2. k-nearest neighbors. The samples x_i and x_j are considered as neighbors if x_i is among the k nearest neighbors of x_j or x_j is among the k nearest neighbors of x_i, where k is a positive integer and the k nearest neighbors are measured by the usual Euclidean distance.

The relations defined by these two methods are both symmetric and consequently the adjacency graph constructed is undirected. The definition of ϵ-ball neighborhood is more geometrically motivated and thus the neighborhood relation reveals the true Euclidean distance among samples. While the ϵ-ball neighborhood method cannot guarantee the connectivity of the whole graph and often leads to several separated subgraphs. On the other hand, the k nearest neighbor approach is easier to obtain a connected graph, while the neighbors defined often cannot well characterize the real geometrical relations among samples. Note that the adjacency structure of the graph is already fixed during the first step and the consequent graph weight calculation step will be constrained by these neighborhood relations.

For graph weight calculation, there exist three frequently used approaches:

1. Heat Kernel [2]:

$$W_{ij} = \begin{cases} e^{-\frac{\|x_i - x_j\|^2}{t}}, & \text{if } x_i \text{ and } x_j \text{ are neighbors,} \\ 0, & \text{otherwise,} \end{cases}$$

   where t is the the heat kernel parameter. Note when $t \to \infty$, the heat kernel will produce binary weight and the graph constructed will be a binary graph. *i.e.*,

$$W_{ij} = \begin{cases} 1, & \text{if } x_i \text{ and } x_j \text{ are neighbors.} \\ 0, & \text{otherwise.} \end{cases}$$

2. Inverse Euclidean Distance [6]: $W_{ij} = \|x_i - x_j\|^{-1}$, where $\|\cdot\|$ is the Euclidean distance.

3. Local Linear Reconstruction Coefficient [13]: Roweis et al. proposed to utilize the local linear reconstruction coefficients as graph weights. The motivation is to reconstruct the sample from its neighboring points and minimize the ℓ_2 reconstruction error defined as

$$\xi(W) = \sum_i \|x_i - \sum_j W_{ij}x_j\|^2, s.t. \sum_j W_{ij} = 1, \; \forall\, i, \quad (2.1)$$

   where $W_{ij} = 0$ if sample x_i and x_j are not neighbors.

3 ℓ_1 Graph: Motivation and Construction

3.1 Motivation: Compact vs. Sparse Representation

Principal Component Analysis (PCA) [3] has been widely used for dimensionality reduction and feature representation. The algorithm encodes the input data as reduced dimensional vectors by following the criteria of minimal mean square reconstruction error. In another perspective, it seeks a transform that preserves maximal energy in the representation. Denote the linear transformation matrix as P, we have

$$\hat{P} = \arg \max_{P^T P = I} \frac{1}{N} \sum_{i=1}^{N} \|P^T(x_i - \bar{x})\|^2, \quad (3.2)$$

where $\|\cdot\|^2$ is the Frobenius norm of a matrix, $\bar{x}$ is the average vector of all samples, and I is the identity matrix. Equation (3.2) can be further expressed as

$$\begin{aligned} \hat{P} &= \arg \max_{P^T P = I} \frac{1}{N} Tr(P^T \sum_i (x_i - \bar{x})(x_i - \bar{x})^T P) \\ &= \arg \max_{P^T P = I} Tr(P^T Cov P), \quad (3.3) \end{aligned}$$

where $Tr(\cdot)$ denotes the trace of a square matrix and $Cov = \frac{1}{N}\sum_i (x_i - \bar{x})(x_i - \bar{x})^T$ is the covariance matrix of all the training samples. The optimization of the objective can be solved by the generalized eigenvalue decomposition method [3]. Note the energy to be maximized or the square error to be minimized in PCA is in the form of ℓ_2 distance. The code/representation derived by PCA is called *compact* code [7].

On the other hand, it has been proposed [7] that compact coding scheme is insufficient to account for the receptive field properties of cells in the mammalian visual pathway. In contrast, the human vision system is near to optimality in the representation of natural scenes only if optimality is defined in terms of sparsely distributed coding rather than the compact coding.

Sparse coding describes the image in terms of a linear superposition of basis functions plus the noise [12], namely,

$$y = \sum_i a_i \phi_i + e, \quad (3.4)$$

where y is the sample image to be represented, ϕ_i is the ith basis function and e is the noise or the reconstruction error.

Olshausen et al. [10] employ Bayesian models and impose priors to the coefficients a_i for deducing the sparse representation. Wright et al. [16] directly use the training images as basis functions and design an ℓ_1 optimization algorithm for the image representation. Arrange the coefficients a_i to form the coefficient vector a, and we have,

$$\hat{a} = \arg \min_a \|y - Xa\|_1, \quad (3.5)$$

where $\|\cdot\|_1$ is the ℓ_1 norm.

The advantage of the algorithms based on ℓ_1 reconstruction error minimization is that the selection of basis functions and the calculation of the coefficients are conducted simultaneously and no assumption is required on the coefficient distribution. Although no sparse priors are imposed, the sparse property of the coefficient vector a is generated naturally by the ℓ_1 optimization when the system is underdetermined.

3.2 ℓ_1 Graph Construction In traditional graph construction process, the graph adjacency structure and the graph weights are derived separately. We argue that the graph adjacency structure and the graph weights are interrelated and should not be separated. Thus it is desired to develop a procedure which can simultaneously completes these two tasks within one step.

The coefficients deduced in the ℓ_1 optimization essentially characterize the relation among image samples and thus it is natural to utilize the ℓ_1 optimization for the graph construction. Moreover, the ℓ_1 optimization process automatically produces a sparse representation, which can be employed naturally as the indication of the graph adjacency structure. Consequently the graph adjacency structure and graph weights are determined simultaneously.

Equation (3.5) essentially minimizes the ℓ_1 norm of the reconstruction error, or the residual $e = y - Xa$. The well-poseness of the minimization relies on the condition $d \ll N$, *i.e.*, the sample number is much larger than the feature dimension. As it is often the case that the sample number is smaller than the feature dimension in vision related problem, the system becomes underdetermined and consequently the coefficient a is no longer sparse. To overcome this issue, Wright et al. [16] proposed to minimize both the reconstruction error and the coefficient norm, and turned to solve

$$\hat{a'} = \arg\min \|a'\|_1, \ s.t. \ Ba' = y, \tag{3.6}$$

where $B = [X, I] \in \mathbb{R}^{d \times (d+N)}$. The coefficients of some image samples are demonstrated in Figure 2.

For each sample image, we regard all the other samples in the training set as basis functions and implement the ℓ_1 minimization process in (3.6) to obtain the sparse representation. The coefficients for the ℓ_1 reconstruction reflect the relation among samples, and hence the graph adjacency structure as well as the graph weights of the ℓ_1 graph is defined as,

Graph Adjacency Structure: A directed edge is placed from node j to i iff $a^i_j \neq 0$, where a^i_j is the coefficient corresponding to the jth sample basis function in the representation of sample x_i.

Graph Weights: $W_{ij} = |a^i_j|$.

The reconstruction coefficients in the sparse sample representation essentially reflect a close relation between the image pairs, and the amplitude of the corresponding coefficient naturally weighs the relation. As a consequence, we use the absolute value of the coefficients a^i_j as the ℓ_1 graph weights.

Note here the constructed ℓ_1 graph is a directed graph, and the detailed procedure for ℓ_1 graph construction is listed in Algorithm 1. The semi-supervised learning framework based on the directed ℓ_1 graph will be presented in the next section.

3.3 ℓ_1 versus ℓ_2 The LLE [13] framework also tries to minimize the linear reconstruction error. Here we would like to highlight the differences between the ℓ_1 graph and the ℓ_2 LLE graph:

1. The ℓ_2 minimization in the traditional LLE algorithm does not lead to a sparse representation due to the property of the ℓ_2 isocontour [16].
2. The reconstruction and minimization of traditional LLE is only processed within the sample neighbors defined by the k nearest neighbor or the ϵ-neighbor graph methods. The structure of the graph adjacency has been determined by the previous step and LLE algorithm only produces the graph weights. On the other hand, the optimization of the ℓ_1 algorithm is carried out using all the sample images. The sparseness property of the deduced ℓ_1 coefficients is naturally used to express both graph adjacency structure and the graph weights.
3. No parameter is required for the ℓ_1 graph construction while for the LLE algorithm, the parameter k or ϵ must be set manually, and the optimal setting may be different for different data sets.

Algorithm 1 ℓ_1 Directed Graph Construction

```
Input: Column sample matrix $X = [x_1, x_2, \ldots, x_N]$.
Normalize the training samples to have unit $\ell_2$ norm.
For $i = 1 : N$, Do
    • Set $X \setminus x_k = [x_1, x_2, \ldots, x_{k-1}, x_{k+1}, \ldots, x_N]$, and $B = [X \setminus x_k, I]$;
    • Solve the $\ell_1$ minimization problem,$\hat{a'} = \arg\min \|a'\|_1, \ s.t. \ Ba' = x_k$,
      and we get $a = a'_{1:N}$, $e = a'_{N+1:N+d}$.
    • For $j = 1 : N - 1$
        If $j < i$    Set $W_{ij} = |a_j|$;
        else          Set $W_{ij} = |a_{j-1}|$.
      end
end
Output $W$
```

4 Semi-supervised Learning over ℓ_1 Graph

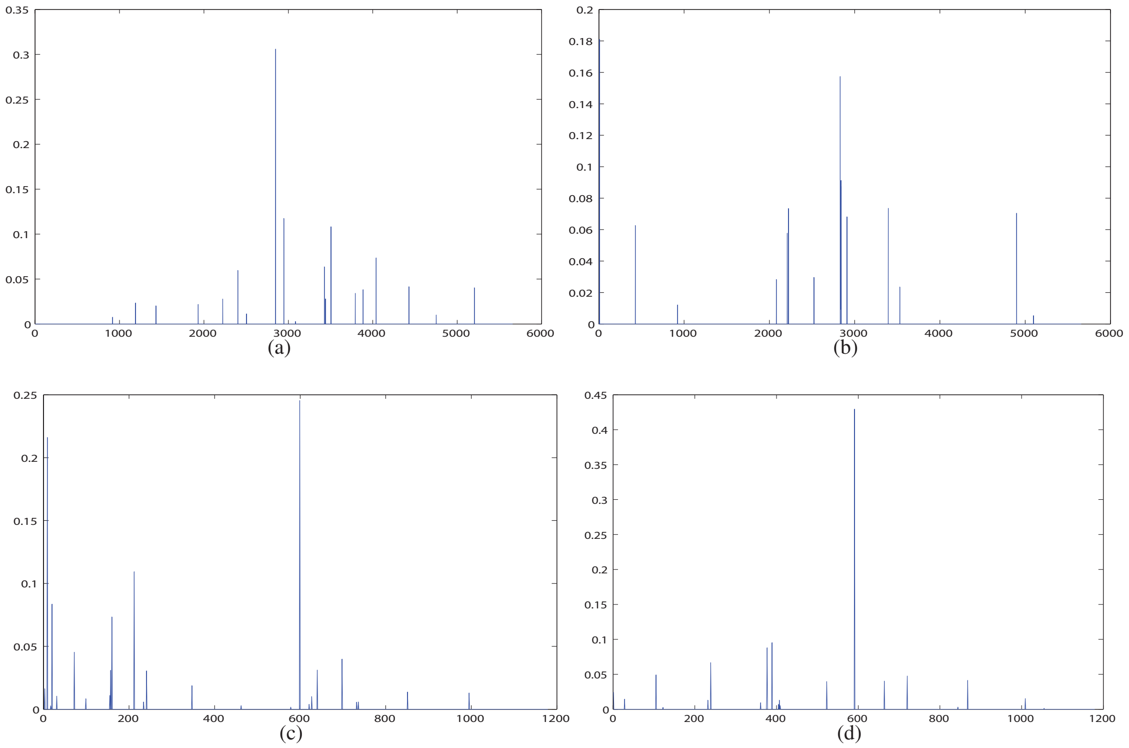
The ℓ_1 graph gathers the relation expressed by sparse sample representation and the adjacency matrix W is asymmetric. Similar to [20], the objective of semi-supervised learning framework with ℓ_1 graph is based on the so called graph preserving criteria [17],

$$\min E(Y) = \sum_{i,j} W_{ij} \|y_i - y_j\|^2, \tag{4.7}$$

where $Y = [y_1, y_2, \cdots, y_N]$ and y_i is a vector characterizing the probabilities of the sample x_i belonging to different classes, namely,

$$y_i(k) = p(k|x_i), \ k = 1, 2, \cdots, K, \tag{4.8}$$

where K is the total class number, and $p(k|x_i)$ is the posterior probability of the class k for the given sample x_i. Under the semi-supervised learning configuration, we denote $Y = [Y_l, \ Y_u]$ where Y_l includes the class probability vectors for the samples with labels and Y_u includes the class probability vectors for the samples without labels. For a



(a) (b) (c) (d)

Figure 2: Sparse reconstruction coefficients for different samples. (a)(b) FRGC database, and (c)(d) XM2VTS database. The figures also show the adaptively selected neighbor numbers for different samples.

labeled datum x_i, $y_i(k) = 1$ if x_i belongs to the k-th class; $y_i(k) = 0$, otherwise.

An explanation of this objective is as follows. When the samples x_i and x_j are similar, namely, the graph weight W_{ij} is large, the distance between y_i and y_j should be small in order to minimize the objective, namely the class information of the sample x_i and x_j should be similar. Therefore the similarity of the ℓ_1 graph is preserved in label propagation. The objective can be further rewritten as,

$$\begin{aligned} E(Y) &= \sum_i y_i^T y_i D_{ii} + \sum_j y_j^T y_j D^*_{jj} - \sum_{ij} (y_i^T y_j + y_j^T y_i) W_{ij} \\ &= Tr(YDY^T + YD^*Y^T - Y(W + W^T)Y^T) \\ &= Tr(Y(L + L^*)Y^T) = Tr(YCY^T), \end{aligned} \tag{4.9}$$

where D is a diagonal matrix whose diagonal elements are the row sums of the corresponding rows of W, and D^* is the a diagonal matrix whose diagonal elements are the column sums of the corresponding columns of W. $L = D - W$ and $L^* = D^* - W^T$ are the row and column Graph Laplacian matrices respectively.

Note here L and L^* are generally asymmetric for ℓ_1 graph, while $C = L + L^*$ is a symmetric matrix. The symmetric property of the matrix C is guaranteed by the following lemma:

Lemma-1 The constant matrix C in the objective (4.9) is symmetric.

Proof Let

$$W^\sharp = W + W^T, \tag{4.10}$$

then

$$L^\sharp = D^\sharp - W^\sharp = F_D(W^\sharp) - W^\sharp, \tag{4.11}$$

where $F_D(\cdot)$ is an operator to derive a diagonal matrix whose diagonal elements are the row sums of the input matrices.

Replacing $W^\sharp$ in Equation (4.11) with the $W^\sharp$ defined in Equation (4.10), we then have

$$\begin{aligned} L^\sharp &= F_D(W + W^T) - (W + W^T) \\ &= D - W + D^* - W^T = L + L^* = C. \end{aligned} \tag{4.12}$$

Thus the constant matrix C in the objective (4.9) is the Graph Laplacian of the graph with the weight matrix $W^\sharp$. Note $W^\sharp$ is symmetric and thus $C = L^\sharp$ is a symmetric matrix. □

Take derivative of $E(Y)$ with respect to Y, then we have

$$YC = 0, \tag{4.13}$$

that is,

$$\left(\begin{array}{cc} Y_l & Y_u \end{array} \right) \left(\begin{array}{cc} C_{ll} & C_{lu} \\ C_{ul} & C_{uu} \end{array} \right) = 0. \tag{4.14}$$

Finally we get the matrix form relation between the labeled and unlabel samples as

$$y_u = -Y_l C_{lu} C_{uu}^{-1}. \tag{4.15}$$

As a special case, when the graph weight matrix W is symmetric, Equation (4.13) becomes $YL = 0$. By following a similar deduction, we have

$$Y_u = -Y_l L_{lu} L_{uu}^{-1}. \tag{4.16}$$

which is equivalent to the result in [20] when y_i is only a value instead of a vector.

Note that semi-supervised learning based on graph can also be integrated with feature extraction process, and the representative work is the Semi-supervised Discriminant Analysis (SDA) proposed by Cai et al. [5]. SDA combines the smoothness regularization term defined in the low dimensional feature space with the intra-class scatter, and then the generalized eigenvalue decomposition method is applied to seek the projection matrix for dimensionality reduction. Finally the classification is conducted with Nearest Neighbor method in the derived low dimensional feature space. SDA is a general algorithm and can be integrated with any graph, and hence our proposed ℓ_1 graph can be also integrated with SDA to further boost the algorithmic classification capability.

5 Experiments

In this section we systematically evaluate the effectiveness of our proposed semi-supervised learning framework based on ℓ_1 graph in three aspects. First we visualize the sparse graph constructed from the real-world databases. Then the semi-supervised learning based on ℓ_1 graph is carried out on six benchmark facial databases and one general object database in comparison with other popular graphs for the semi-supervised learning problem. The semi-supervised learning is conducted with two configurations. One is directly conducted on the original feature space, namely in the way as discussed in Section 4 and [20]. The other is combined with dimensionality reduction process, and the Semi-supervised Discriminant Analysis (SDA) [5] is used to evaluate the performances of different graphs in utilizing unlabeled data for enhancing classification accuracy. Finally we examine the robustness of the algorithms and report the performance w.r.t. the variation of the number of labeled samples.

5.1 Date Sets Preparation We gathered almost all the popular face databases for our experiments. Six data sets are used, including Face Recognition Grand Challenge database (FRGC version 1.0) [11], YALE, FERET, ORL, CMU PIE and XM2VTS databases [1]. For all these databases, facial images are aligned by fixing the locations of two eyes. The XM2VTS database contains 1180 photos taken from 295 persons with 4 photos for each person. The images are manually cropped and normalized to the size of 36-by-32 pixels. The ORL database contains 400 images of 40 persons, where each image is manually cropped and normalized to the size of 32-by-28 pixels. For the left four databases, facial images are normalized to the size of 32-by-32 pixels. We use the seventy people with six images for each person for the FERET database. The CMU PIE (Pose, Illumination, and Expression) database contains more than $40,000$ facial images of 68 people. In our experiment, a subset of five near frontal poses (C27, C05, C29, C09 and C07) and illuminations indexed as 08 and 11 are used for the face recognition experiments. The Yale face database contains 165 grayscale images of 15 individuals with 11 images per subject, one per different facial expression or configuration: center-light, with/without glasses, happy, left-light, normal, right-light, sad, sleepy, surprised, and wink. The FRGC database consists of 5658 images of 275 subjects, and the number of facial images of each subject varies from 6 to 48.

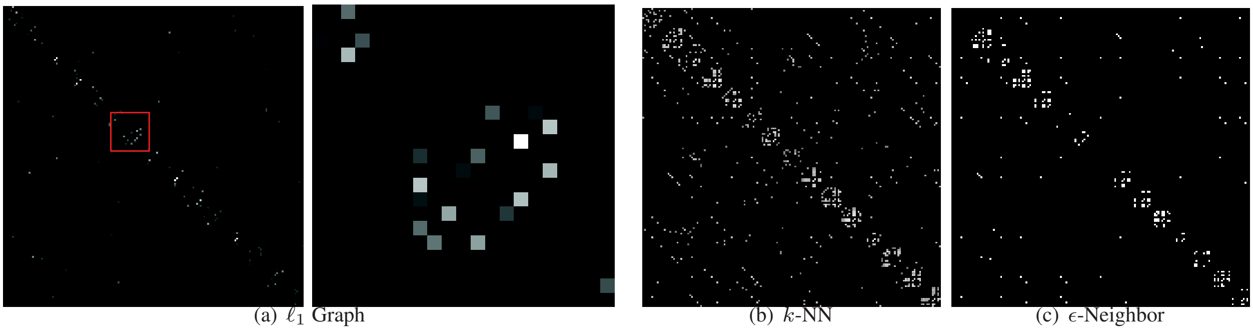
We also include a general object recognition data set, i.e., the ETH-80 database[2], which contains 3280 images taken from 80 generic subjects with each subject 41 photos. The subjects belong to 8 different classes with 10 subjects for each class. Again the object images are normalized to the size of 32-by-32 in our experiment.

5.2 Visualization of the Sparse Graph Adjacency Matrix In this subsection, we demonstrate the visual property of the sparse graph weight matrix W in comparison to the traditional graphs. We construct the ℓ_1 sparse graph, the k-nearest neighbor graph, and the ϵ-neighbor graph on the YALE face database. The graph adjacency matrices are demonstrated in Fig 3. From this figure, we can have two observations: a) The edges in the ℓ_1 graph are very sparse; and 2) there are much less inter-subject adjacency connections in the ℓ_1 graph than other graphs, which means that ℓ_1 graph encodes more discriminative information and hence is more effective in guiding the propagation of the class labels than other traditional graphs.

5.3 Classification Accuracy

[1] Available at http://www.face-rec.org/databases/

[2] Available at http://www.vision.ethz.ch/projects/categorization/



(a) ℓ_1 Graph (b) k-NN (c) ϵ-Neighbor

Figure 3: Visualization of the graph adjacency matrix of (a) sparse ℓ_1 graph (the right image is a zoom in of the area indicated by the red square in the left image), (b) k-nearest neighbor graph (k=3), and (c) ϵ-neighbor graph (ϵ=1300) on YALE database, where images from the same subject are arranged together.

5.3.1 Parameter-free label propagation with ℓ_1 graph

In this subsection we carry out the classification experiments on the face databases using our proposed graph based semi-supervised label propagation algorithm. The ℓ_1 sparse graph is compared to the traditional graphs in the semi-supervised learning configuration. For all the six face databases, we randomly sampled different percentages of images as the labeled samples, and the left are used as unlabeled data for semi-supervised learning. No parameter is required for the ℓ_1 graph based algorithm, and for the other graphs, we vary the values of k and ϵ in the traditional graph construction method and report the classification results for different configurations with $k = 3, 6$ and $\epsilon = 2000, 4000$, which are comparably robust across different databases. The performance w.r.t. the variation of different k and ϵ is evaluated in the following sections. For the graph weights, the gaussian kernel $K(x, y) = \exp\{-\|x - y\|^2/\delta_o^2\}$ with parameters δ_o set as $2^{1/2.5}\delta$ empirically, where δ is the standard deviation of the sample data. We also compare the ℓ_1 graph with the ℓ_2 graph used in LLE algorithm. For the ℓ_1 graph and the LLE graph, the graph adjacency matrix W is asymmetric and a symmetrization process is used as in Equation (4.10). Equation (4.15) is used to derive the class labels for those unlabeled samples. The k-NN graph and the ϵ-neighbor graph have symmetric W, and hence Equation (4.16) is employed as a special case of Equation (4.15) for label inference.

The classification error rates for semi-supervised leaning based on different graphs are shown in Table 1, from which we can have a set of observations:

1. The ℓ_1 graph generally achieves a highest recognition accuracy compared to those traditional graphs, followed by the LLE graph.

2. The performance of the k-NN graph is consistently better than the ϵ-neighbor graph. In our experiments, we found it extremely difficult to find a proper ϵ for the ϵ-neighbor graph. The k-nearest neighbor graph is generally more robust than the ϵ graph.

3. The algorithm works well especially when there exist enough labeled data for the training set. No parameter tuning is required in our proposed framework and the algorithm can adjust to different data sets automatically.

5.3.2 Semi-supervised Discriminant Analysis with ℓ_1 graph

To further examine the effectiveness of the proposed ℓ_1 graph, we also conduct experiments on face databases of ORL, PIE and FERET using the semi-supervised discriminant analysis algorithm (SDA) introduced in [5]. First we utilize different approaches in the graph construction process, and then we implement the SDA algorithm for dimensionality reduction. Finally the nearest neighbor approach is employed for the final classification in the derived low dimensional feature subspace. Similar graph symmetrization process, i.e., $W^\sharp = (W + W^T)$, is used for the ℓ_1 graph and the LLE graph in order to satisfy the algorithmic requirements of SDA. For all the algorithms compared, we search over all the possible dimensions and report the best performance as conventionally. The detailed face recognition accuracies are shown in Table 2, from which we can see: 1) the performance of ℓ_1 graph is also robust with the SDA dimensionality reduction algorithm and performs better than or is comparable with other graphs with best parameters; and 2) the LLE graph shows also very good in this scenario, but the best performance often corresponds to different algorithmic parameter for different experimental configuration.

5.4 Parameter Sensitivity of Traditional Graphs

In this subsection we examine the parameter sensitivity of traditional graphs. We vary the graph parameters in traditional graph, i.e., k and ϵ, and examine the recognition performance

Table 1: Recognition error rates (%) of different graphs using the prosed label propagation algorithms (denoted as Semi-Prop) on the seven databases with different configurations. Note that the bold numbers are the best accuracies for each configuration and the percentage number after the data set name is the percentage of the labeled samples.

Semi-Prop	ℓ_1 Graph	k-NN Graph		ϵ Graph		LLE Graph	
Dataset(label%)	N/A	k=3	k=6	ϵ=2000	ϵ=4000	k=3	k=6
YALE(50%)	**29.3**	32.0	36.0	37.33	38.67	32.0	32.0
YALE(60%)	**21.7**	33.3	33.3	40.0	41.7	26.7	26.7
YALE(80%)	**10.0**	23.3	13.3	16.7	20.0	26.7	13.3
ORL(50%)	**6.0**	13.5	19.5	45.5	32.5	10.0	7.5
ORL(60%)	**5.0**	13.1	17.5	41.3	31.9	9.4	9.4
ORL(80%)	**3.8**	7.5	13.8	37.5	32.5	**3.8**	5.0
PIE(50%)	**14.0**	20.6	21.3	35.9	25.7	19.7	26.7
PIE(60%)	**9.1**	15.9	18.7	26.6	22.6	14.3	16.7
PIE(80%)	**1.6**	15.1	15.9	16.7	7.1	2.4	4.8
FERET(50%)	**11.8**	23.8	34.3	50.0	39.1	16.2	18.1
FERET(60%)	**7.1**	22.9	32.9	49.3	35.7	10.0	8.6
FERET(80%)	**1.4**	22.9	42.9	52.9	31.4	5.7	2.9
XM2VTS(50%)	**4.6**	15.6	24.8	45.4	33.2	12.5	11.7
XM2VTS(60%)	**5.4**	16.6	22.5	46.6	32.5	12.0	11.0
XM2VTS(80%)	**2.0**	10.2	14.9	24.4	17.0	7.5	5.4
FRGC(50%)	**22.3**	41.6	47.8	84.1	94.7	36.6	35.0
ETH(50%)	**6.6**	11.2	13.0	60.9	59.8	10.4	9.5

on the FERET database in the evaluation. The results are shown in Figure 4 (b). We can see that the recognition accuracy is dramatically influenced by the graph parameters, especially for the k-NN and ϵ graph. No graph parameter is required for the ℓ_1 graph and thus the accuracy of ℓ_1 graph remains the same in the evaluation.

5.5 Influence of the Label Number In this subsection we evaluate the influence of the label number. FERET database is used in this evaluation. We vary the percentage of the number of labeled samples in this test and draw the recognition error rates over the percentage of labels using the proposed semi-supervised learning algorithm. The performance is shown in Figure 4 (a), from which we observe that graphs constructed by reconstruction methods, no matter it is ℓ_1 or ℓ_2, consistently outperform the k-NN graph and ϵ graph, and ℓ_1 graph is robust to the label percentage variations.

6 Conclusions and Future Work

In this paper, we have presented a parameter-free way to construct graph for semi-supervised learning problems. The underlying philosophy is that each datum can be sparsely encoded as the linear combination of all other data by solving an ℓ_1 optimization problem. In this way, the graph adjacency structure and the graph weights are derived simultaneously in a parameter-free manner. This so called ℓ_1 graph coincides with the human vision system in the representation of natural scenes by sparse coding. Extensive experiments on both face recognition and image classification well validated the superiority of the ℓ_1 graph for semi-supervised learning over other traditional graphs. The proposed ℓ_1 graph opens a new direction for dimensionality reduction research, and we are planning to further study the ℓ_1 graph in several aspects: 1) to use ℓ_1 graph for unsupervised image clustering and categorization; 2) to use ℓ_1 graph for semi-supervised regression problems; 3) to use ℓ_1 graph for directly designing dimensionality reduction algorithm by following the Graph Embedding framework as proposed in [17].

Acknowledgment

This work is supported by NRF/IDM Grant of R-263-000-524-279, Singapore.

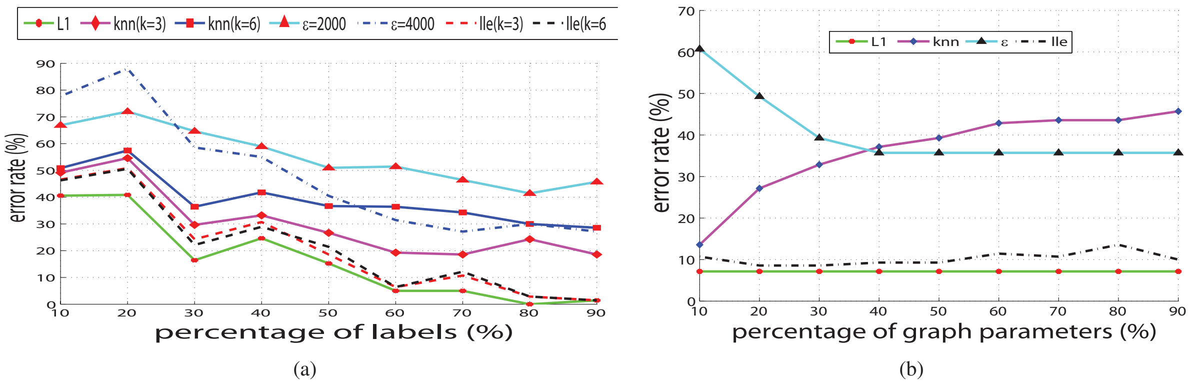
References

[1] P. Belhumeur, J. Hespanda, and D. Kiregeman. Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. *TPAMI*, vol. 19, pp. 711–720, 1997.

[2] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimen-

Table 2: Recognition error rates (%) of different graphs using the Semi-supervised Discriminant Analysis algorithm on the ORL, PIE and FERET databases with different configurations. The bold numbers are the best accuracies for each configuration and the percentage number after the data set name is the percentage of the labeled samples.

SDA	ℓ_1 Graph	k-NN Graph		ϵ Graph		LLE Graph	
Dataset(label%)	N/A	k=3	k=6	ϵ=2000	ϵ=4000	k=3	k=6
ORL(50%)	**7.0**	13.0	19.5	71.0	94.0	9.5	8.0
ORL(60%)	**5.6**	11.9	18.1	76.3	93.1	8.1	8.8
ORL(80%)	**3.8**	8.8	12.5	76.3	96.3	**3.8**	5.0
PIE(50%)	**4.8**	5.1	7.9	27.3	41.0	**4.8**	**4.8**
PIE(60%)	**7.9**	**7.9**	**7.9**	21.4	34.9	**7.9**	**7.9**
PIE(80%)	**1.6**	**1.6**	**1.6**	5.6	13.5	**1.6**	**1.6**
FERET(50%)	**15.7**	24.8	41.4	70.0	97.1	**15.7**	17.1
FERET(60%)	**8.6**	22.9	40.0	62.1	95.7	9.3	**8.6**
FERET(80%)	**2.9**	28.6	41.4	61.4	92.9	5.7	**2.9**



(a)

(b)

Figure 4: (a) Classification error rate v.s. label percentage on FERET database. (b) Recognition error rate w.r.t. the variation of graph parameters on FERET database with 60% samples labeled. The x axis is the ratio between the graph parameter k or ϵ and the maximum value with k_{max}=20 and ϵ_{max}=10000.

sionality reduction and data representation. Neural Computation, vol. 15, no. 6, pp. 1373-1396, 2003.

[3] I. Joliffe. Principal component analysis. *Springer-Verlag, New York*, 1986.

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

[5] D. Cai, X. He, and J. Han. Semi-supervised discriminant analysis. ICCV, 2007.

[6] C. Cortes, and M. Mohri. On transductive regression. NIPS, 2007.

[7] D. Field. What is the goal of sensory coding? Neural Computation, 1994.

[8] Z. Guo, Z. Zhang, E. Xing, C. Faloutsos: Semi-Supervised Learning Based on Semiparametric Regularization. SDM, pp. 132-142, 2008.

[9] N. Meinshausen and P. Bühlmann. *High-dimensional Graphs and Variable Selection with the Lasso*. Annals of Statistics, 34(3):1436-1462, 2006.

[10] B. Olshausen and D. Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? Vision Research, 1997.

[11] P. Phillips, P. Flynn, T. Scruggs, K. Bowyer, K. Chang, J. Marques, J. Min, and W. Worek. Overview of the face recognition grand challenge. CVPR, 2005.

[12] R. Rao, B. Olshausen, and M. Lewicki. Probabilistic Models of the Brain: Perception and Neural Function. MIT Press, 2002.

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

[14] C. Schellewald and C. Schnorr. Probabilistic Subgraph Matching Approach Based on Convex Relaxation. EMMCVPR, 2005.

[15] J. Tenenbaum, V. Silva, and J. Langford. A global geometric framework for nonlinear dimensionality reduction. Science, vol. 290. no. 5500, pp. 2319-2323, 2000.

[16] J. Wright, A. Ganesh, A. Yang, and Y. Ma: Robust face recognition via sparse representation. TPAMI, in press, 2008.

[17] S. Yan, D. Xu, B. Zhang, Q. Yang, H. Zhang, and S. Lin. Graph embedding and extensions: A general framework for dimensionality reduction. TPAMI, vol. 29, no. 1, pp. 40-51, 2007.

[18] D. Zhang, J. Wang, F. Wang, and C. Zhang: Semi-Supervised Classification with Universum. SDM, pp. 323-333, 2008.

[19] D. Zhou, O. Bousquet, T. Lal, J. Weston, and B. Schoelkopf. Learning with local and global consistency. NIPS, 2004.

[20] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using Gaussian fields and harmonic functions. ICML, 2003.