

Learning With ℓ^1 -Graph for Image Analysis

Bin Cheng, Jianchao Yang, *Student Member, IEEE*, Shuicheng Yan, *Senior Member, IEEE*, Yun Fu, *Member, IEEE*, and Thomas S. Huang, *Life Fellow, IEEE*

Abstract—The graph construction procedure essentially determines the potentials of those graph-oriented learning algorithms for image analysis. In this paper, we propose a process to build the so-called directed ℓ^1 -graph, in which the vertices involve all the samples and the ingoing edge weights to each vertex describe its ℓ^1 -norm driven reconstruction from the remaining samples and the noise. Then, a series of new algorithms for various machine learning tasks, e.g., data clustering, subspace learning, and semi-supervised learning, are derived upon the ℓ^1 -graphs. Compared with the conventional k -nearest-neighbor graph and ϵ -ball graph, the ℓ^1 -graph possesses the advantages: 1) greater robustness to data noise, 2) automatic sparsity, and 3) adaptive neighborhood for individual datum. Extensive experiments on three real-world datasets show the consistent superiority of ℓ^1 -graph over those classic graphs in data clustering, subspace learning, and semi-supervised learning tasks.

Index Terms—Graph embedding, semi-supervised learning, sparse representation, spectral clustering, subspace learning.

I. INTRODUCTION

AN informative graph, directed or undirected, is critical for those graph-oriented algorithms designed for the purposes of data clustering, subspace learning, and semi-supervised learning. Data clustering often starts with a pairwise similarity graph and is then transformed into a graph partition problem [17]. The pioneering works on manifold learning, e.g., ISOMAP [18], locally linear embedding [16], and Laplacian Eigenmaps [5], all rely on graphs constructed in different ways. Moreover, most popular subspace learning algorithms, e.g., principal component analysis [12], linear discriminant analysis [3], and locality preserving projections [10], can all be explained within the graph embedding framework as claimed in [20]. Also, most semi-supervised learning algorithms are driven by certain graphs constructed over both labeled and unlabeled data. Zhu *et al.* [22] utilized the harmonic property

Manuscript received May 24, 2009; revised October 19, 2009. First published December 22, 2009; current version published March 17, 2010. This work was supported in part by the NRF/IDM Program, under research Grant NRF2008IDM-IDM004-029, and in part by the U.S. government VACE Program. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Mark (Hong-Yuan) Liao.

B. Cheng and S. Yan are with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore (e-mail: chengbin@nus.edu.sg; eleyans@nus.edu.sg).

J. Yang and T. S. Huang are with the Beckman Institute, University of Illinois at Urbana-Champaign, Urbana, IL USA (e-mail: jyang29@ifp.uiuc.edu; huang@ifp.uiuc.edu).

Y. Fu is with the Department of Computer Science and Engineering, University at Buffalo, State University of New York, Buffalo, NY 14260-2000 USA (e-mail: raymondyunfu@gmail.com).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TIP.2009.2038764

of Gaussian random field over the graph for semi-supervised learning. Belkin and Niyogi [4] instead learned a regression function that fits the labels at labeled data and also maintains smoothness over the data manifold expressed by a graph.

There exist two popular ways for graph construction, one of which is the k -nearest-neighbor method, and the other is the ϵ -ball based method, where, for each datum, the samples within its surrounding ϵ ball are connected, and then various approaches, e.g., binary, Gaussian-kernel [5] and ℓ^2 -reconstruction [16], can be used to further set the graph edge weights. Since the ultimate purposes of the constructed graphs are for data clustering, subspace learning, semi-supervised learning, etc., the following graph characteristics are desired.

- 1) **Robustness to data noise.** The data noises are inevitable especially for visual data, and the robustness is a desirable property for a satisfactory graph construction method. The graph constructed by k -nearest-neighbor or ϵ -ball method is founded on pair-wise Euclidean distance, which is very sensitive to data noise. It means that the graph structure is easy to change when unfavorable noises come in.
- 2) **Sparsity.** Recent research on manifold learning [5] shows that sparse graph characterizing locality relations can convey valuable information for classification purpose. Also, for applications with large scale data, a sparse graph is the inevitable choice due to the storage limitation.
- 3) **Datum-adaptive neighborhood.** Another observation is that the data distribution probability may vary greatly at different areas of the feature space, which results in distinctive neighborhood structure for each datum. Both k -nearest-neighbor and ϵ -ball methods, however, use a fixed global parameter to determine the neighborhoods for all the data, and hence fail to offer such datum-adaptive neighborhoods.

We present in Section II a procedure to construct robust and datum-adaptive ℓ^1 -graph by utilizing the overall contextual information instead of only pairwise Euclidean distance as conventionally. The neighboring samples of a datum and the corresponding ingoing edge weights are simultaneously derived by solving an ℓ^1 -norm optimization problem, where each datum is reconstructed by the linear combination of the remaining samples and noise item, with the objective of minimizing the ℓ^1 norm of both reconstruction coefficients and data noise. Compared with the graphs constructed by k -nearest-neighbor and ϵ -ball methods, the ℓ^1 -graph has the following three advantages. First, ℓ^1 -graph is robust owing to the overall contextual ℓ^1 -norm formulation and the explicit consideration of data noises. Fig. 1(a) shows the graph robustness comparison between ℓ^1 -graph and k -nearest-neighbor graph. Second, the sparsity of the ℓ^1 -graph is automatically determined instead of manually as in k -nearest-neighbor and ϵ -ball methods.

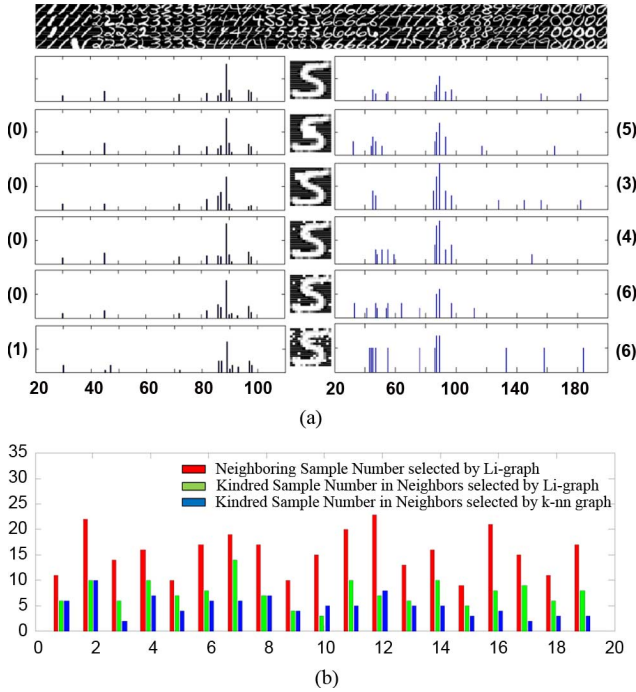


Fig. 1. Robustness and adaptiveness comparison for neighbors selected by ℓ^1 -graph and k -nearest-neighbor graph. (a) Illustration of basis samples (first row), reconstruction coefficient distribution in ℓ^1 -graph (left), samples to reconstruct (middle, with added noises from the third row on), and similarity distribution of the k nearest neighbors selected with Euclidean distance (right) in k -nn graph. Here, the horizontal axes indicate the index number of the training samples. The vertical axes of the left column indicate the reconstruction coefficient distribution for all training samples in sparse coding, and those of right column indicate the similarity value distribution of k nearest neighbors. Note that the number in parenthesis is the number of neighbors changed compared with results in the second row, and ℓ^1 -graph shows much more robust to image noises. (b) Neighboring samples comparison between ℓ^1 -graph and k -nearest-neighbor graph. The red bars indicate the numbers of the neighbors selected by ℓ^1 -graph automatically and adaptively. The green bars indicate the numbers of kindred samples among the k neighbors selected by ℓ^1 -graph, and the blue bar indicate the numbers of kindred samples within the k nearest neighbors measured by Euclidean distance in k -nn graph. Note that the results are obtained on USPS digit database [11] and the horizontal axis indicates the index of the reference sample to reconstruct. (a) Neighbor robustness comparison of ℓ^1 -graph and k -nn graph. (b) Datum-adaptive neighbor numbers selected by sparse ℓ^1 -graph, and kindred neighbor numbers for ℓ^1 -graph and k -nn graph.

Finally, the ℓ^1 -graph is datum-adaptive. As shown in Fig. 1(b), the number of neighbors selected by ℓ^1 -graph is adaptive to each datum, which is valuable for applications with unevenly distributed data.

This ℓ^1 -graph is then utilized in Section III to instantiate a series of graph-oriented algorithms for various machine learning tasks, e.g., data clustering, subspace learning, and semi-supervised learning. Owing to the above three advantages over classical graphs, ℓ^1 -graph brings consistent performance gain in all these tasks as detailed in Section IV.

II. RATIONALES ON ℓ^1 -GRAPH

For a general data clustering or classification problem, the training sample set is assumed being represented as a matrix $X = [x_1, x_2, \dots, x_N], x_i \in \mathbb{R}^m$, where N is the sample number and m is the feature dimension. For supervised learning problems, the class label of the sample x_i is then assumed to be $l_i \in \{1, 2, \dots, N_c\}$, where N_c is the total number of classes.

A. Motivations

The ℓ^1 -graph is motivated by the limitations of classical graph construction methods [5], [16] in robustness to data noise and datum-adaptiveness, and recent advances in sparse coding [8], [14], [19]. Note that a graph construction process includes both sample neighborhood selection and graph edge weight setting, which are assumed in this work to be unsupervised, without harnessing any data label information.

The approaches of k -nearest-neighbor and ϵ -ball are very popular for graph construction in literature. Both of them determine the neighboring samples based on *pairwise* Euclidean distance, which is, however, very sensitive to data noises and one noisy feature may dramatically change the graph structure. Also when the data are not evenly distributed, the k nearest neighbors of a datum may involve faraway inhomogeneous data if the k is set too large, and the ϵ -ball may involve only single isolated datum if ϵ is set too small. Moreover, the optimum of k (or ϵ) is datum-dependent, and one single global parameter may result in unreasonable neighborhood structure for certain datum.

The research on sparse coding or sparse representation has a long history. Recent research shows that sparse coding appears to be biologically plausible as well as empirically effective for image processing and pattern classification [19]. Olshausen *et al.* [15] employed the Bayesian models and imposed ℓ^1 priors for deducing the sparse representation, and Wright *et al.* [19] proposed to use sparse representation for direct face recognition. In this work, beyond the sparse coding for individual test datum, we are interested in the overall behavior of the whole sample set in sparse representation, and then present the general concept of ℓ^1 -graph, followed by its applications in various machine learning tasks, e.g., data clustering, subspace learning, and semi-supervised learning.

B. Robust Sparse Representation

Much interest has been shown in computing linear sparse representation with respect to an overcomplete dictionary of the basis elements. Suppose we have an underdetermined system of linear equations: $x = D\alpha$, where $x \in \mathbb{R}^m$ is the vector to be approximated, $\alpha \in \mathbb{R}^n$ is the vector for unknown reconstruction coefficients, and $D \in \mathbb{R}^{m \times n} (m < n)$ is the overcomplete dictionary with n bases. Generally, a sparse solution is more robust and facilitate the consequent identification of the test sample x . This motivates us to seek the sparsest solution to $x = D\alpha$ by solving the following optimization problem:

$$\min_{\alpha} \|\alpha\|_0, \quad s.t. \quad x = D\alpha \quad (1)$$

where $\|\cdot\|_0$ denotes the ℓ^0 -norm, which counts the number of nonzero entries in a vector. But It is well known that the sparsest representation problem is NP-hard in general case, and difficult even to approximate. However, recent results [8], [19] show that if the solution is sparse enough, the sparse representation can be recovered by the following convex ℓ^1 -norm minimization [8]

$$\min_{\alpha} \|\alpha\|_1, \quad s.t. \quad x = D\alpha. \quad (2)$$

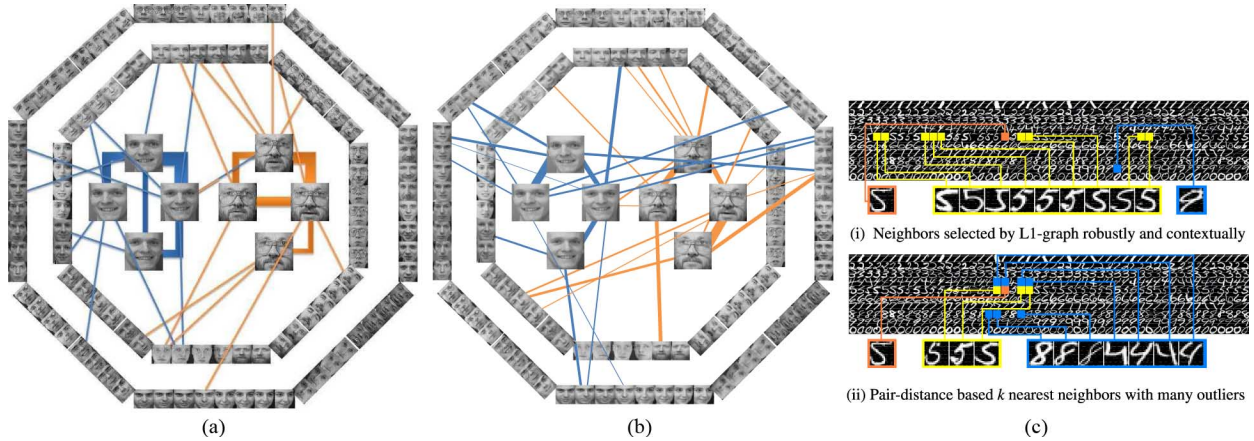


Fig. 2. Visualization comparison of (a) the ℓ^1 -graph and (b) the k -nearest-neighbor graph, where the k for each datum is automatically selected in the ℓ^1 -graph. Note that the thickness of the edge line indicates the value of the edge weight (Gaussian kernel weight for k -nearest-neighbor graph). For ease of display, we only show the graph edges related to the samples from two classes and in total 30 classes from the YALE-B database are used for graph construction. (c) Illustration on the positions of a reference sample (red), its kindred neighbors (yellow), and its inhomogeneous neighbors (blue) selected by (i) ℓ^1 -graph and (ii) k -nearest-neighbor method based on samples from the USPS digit database [11]. (a) Example of ℓ^1 -graph. (b) Example k -nearest-neighbor graph. (c) Example ℓ^1 -graph and fc-NN graph.

This problem can be solved in polynomial time by standard linear programming method [7]. In practice, there may exist noises on certain elements of x , and a natural way to recover these elements and provide a robust estimation of α is to formulate

$$x = D\alpha + \zeta = [D \quad I] \begin{bmatrix} \alpha \\ \zeta \end{bmatrix} \quad (3)$$

where $\zeta \in \mathbb{R}^m$ is the noise term. Then by setting $B = [D \quad I] \in \mathbb{R}^{m \times (m+n)}$ and $\alpha' = \begin{bmatrix} \alpha \\ \zeta \end{bmatrix}$, we can solve the following ℓ^1 -norm minimization problem with respect to both reconstruction coefficients and data noises:

$$\min_{\alpha'} \|\alpha'\|_1, \quad s.t. \quad x = B\alpha'. \quad (4)$$

This optimization problem is convex and can be transformed into a general linear programming problem. There exists a globally optimal solution, and the optimization can be solved efficiently using many available ℓ^1 -norm optimization toolboxes like [2]. Note that the ℓ^1 norm optimization toolbox in [2] may convert the original constrained optimization problem into an unconstrained one, with an extra regularization coefficient which can be tuned for optimum in practice but essentially does not exist in original problem formulation.

C. ℓ^1 -Graph Construction

An ℓ^1 -graph summarizes the overall behavior of the whole sample set in sparse representation. The construction process is formally stated as follows.

- 1) **Inputs:** The sample data set denoted as the matrix $X = [x_1, x_2, \dots, x_N]$, where $x_i \in \mathbb{R}^m$.
- 2) **Robust sparse representation:** For each datum x_i in the sample set, its robust sparse coding is achieved by solving the ℓ^1 -norm optimization problem

$$\min_{\alpha^i} \|\alpha^i\|_1, \quad s.t. \quad x_i = B^i \alpha^i \quad (5)$$

where matrix $B^i = [x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_N, I] \in \mathbb{R}^{m \times (m+N-1)}$ and $\alpha^i \in \mathbb{R}^{m+N-1}$.

- 3) **Graph weight setting:** Denote $G = \{X, W\}$ as the ℓ^1 -graph with the sample set X as graph vertices and W as the graph weight matrix, and we set $W_{ij} = \alpha_j^i$ (nonnegativity constraints may be imposed for α_j^i in optimization if for similarity measurement) if $i > j$, and $W_{ij} = \alpha_{j-1}^i$ if $i < j$.

Fig. 2 depicts partial of the ℓ^1 -graphs based on the data from the YALE-B face database [13] and USPS digit database [11] respectively. An interesting observation from Fig. 2 is that, besides being robust and datum-adaptive, the ℓ^1 -graph has the potential to connect kindred samples, and hence may potentially convey more discriminative information, which is valuable for its later introduced applications in data clustering, subspace learning, and semi-supervised learning. Taking the face image as an example, the intuition behind the observed discriminating power of ℓ^1 graph is that, if one expects to reconstruct a face image with all other face images as bases, the most efficient way in terms of the number of relevant bases is to use similar images or images from the same subject, which leads to a sparse solution and coincides with the empirical observations in [19] for robust face recognition with sparse representation.

1) *Discussions:* 1) Note that the formulation in (4) is based on the assumption that the feature dimension, m , is reasonably large, otherwise the sparsity of noises shall make no sense. It means that (4) is not applicable for simple 2-D or 3-D toy data. 2) In implementation, the data normalization, i.e., $\|x_i\|_2 = 1$, is critical for deriving semantically reasonable coefficients. 3) The k -nearest-neighbor graph is flexible in terms of the selection of similarity/distance measurement, but the optimality is heavily data dependent. In this work, we use the most conventional Euclidean distance for selecting the k nearest neighbors. 4) For certain extreme cases, e.g., if we simply duplicate each sample and generate another new dataset of double size, ℓ^1 -graph may only connect these duplicated pairs, and thus fail to convey valuable information. A good observation is that these extreme cases are very rare for those datasets investigated in general research.

III. LEARNING WITH ℓ^1 -GRAPH

An informative graph is critical for those graph-oriented learning algorithms. Similar to classical graphs constructed by k -nearest-neighbor or ϵ -ball method, ℓ^1 -graph can be integrated with various learning algorithms for various tasks, e.g., data clustering, subspace learning, and semi-supervised learning. In this section, we briefly introduce how to benefit from ℓ^1 -graph for these tasks.

A. Spectral Clustering With ℓ^1 -Graph

Data clustering is the classification of samples into different groups, or more precisely, the partition of samples into subsets, such that the data within each subset are similar to each other. The spectral clustering [17] is among the most popular algorithms for this task, but there exists one parameter δ [17] for controlling the similarity between a data pair. Intuitively the contribution of one sample to the reconstruction of another sample is a good indicator of similarity between these two samples, we decide to use the reconstruction coefficients to constitute the similarity graph for spectral clustering. As the weights of the graph are used to indicate the similarities between different samples, they should be assumed to be non-negative. Using the ℓ^1 -graph, the algorithm can automatically select the neighbors for each datum, and at the same time the similarity matrix is automatically derived from the calculation of these sparse representations. The detailed spectral clustering algorithm based on ℓ^1 -graph is listed as follows.

- 1) Symmetrize the graph similarity matrix by setting the matrix $W = (W + W^T)/2$.
- 2) Set the graph Laplacian matrix $L = D^{-1/2}WD^{-1/2}$, where $D = [d_{ij}]$ is a diagonal matrix with $d_{ii} = \sum_j w_{ij}$.
- 3) Find c_1, c_2, \dots, c_K , the eigenvectors of L corresponding to the K largest eigenvalues, and form the matrix $C = [c_1, c_2, \dots, c_K]$ by stacking the eigenvectors in columns.
- 4) Treat each row of C as a point in \mathbb{R}^K , and cluster them into K clusters via the K -means method.
- 5) Finally, assign x_i to the cluster j if the i th row of the matrix C is assigned to the cluster j .

B. Subspace Learning With ℓ^1 -Graph

Similar to the graph construction process in Locally Linear Embedding (LLE), the ℓ^1 -graph characterizes the neighborhood reconstruction relationship. In LLE, the graph is constructed by reconstructing each datum with its k nearest neighbors or the samples within the ϵ -ball based on the ℓ^2 -norm. LLE and its linear extension, called neighborhood preserving embedding (NPE) [9], both rely on the global graph parameter (k or ϵ). Following the idea of NPE algorithm, ℓ^1 -graph can be used to develop a subspace learning algorithm as follows.

The general purpose of subspace learning is to search for a transformation matrix $P \in \mathbb{R}^{m \times d}$ (usually $d \ll m$) for transforming the original high-dimensional datum into another low-dimensional one. ℓ^1 -graph uncovers the underlying sparse reconstruction relationship of each datum, and it is desirable

to preserve these reconstruction relationships in the dimensionality reduced feature space. Note that in the dimension reduced feature space, the reconstruction capability is measured by ℓ^2 norm instead of ℓ^1 norm for computational efficiency. Then the pursue of the transformation matrix can be formulated as the optimization

$$\min_{P^T X X^T P = I} \sum_{i=1}^N \left\| P^T x_i - \sum_{j=1}^N W_{ij} P^T x_j \right\|^2 \quad (6)$$

where W_{ij} is determined by the constructed ℓ^1 -graph. This optimization problem can be solved with generalized eigenvalue decomposition approach as

$$X M X^T p_{m+1-j} = \lambda_j X X^T p_{m+1-j} \quad (7)$$

where $M = (I - W)^T(I - W)$, and p_{m+1-j} is the eigenvector corresponding to the j th largest eigenvalue λ_j as well as the $(m + 1 - j)$ th column vector of the matrix P .

The derived transformation matrix is then used for dimensionality reduction as

$$y_i = P^T x_i \quad (8)$$

where y_i is the corresponding low-dimensional representation of the sample x_i and finally the classification process is performed in this low-dimensional feature space with reduced computational cost.

C. Semi-Supervised Learning With ℓ^1 -Graph

As shown in Figs. 1 and 2, the ℓ^1 -graph is robust to data noises and datum-adaptive, also empirically has the potential to convey more discriminative information compared with conventional graphs based on k -nearest-neighbor or ϵ -ball method. These properties make ℓ^1 -graph a good candidate for propagating the label information over the graph. Semi-supervised learning recently has attracted much attention, and was widely used for both regression and classification purposes. The main idea of semi-supervised learning is to utilize unlabeled data for improving the classification and generalization capability on the testing data. Commonly the unlabeled data are used as an extra regularization term to the objective functions from traditional supervised learning algorithms.

In this work, the unlabeled data are used to enlarge the vertex number of the ℓ^1 -graph, and further enhance the robustness of the graph. Finally the ℓ^1 -graph based on both labeled and unlabeled data is used to develop semi-supervised learning algorithm. Here, we take marginal Fisher analysis (MFA) [20] as an example for the supervised part in semi-supervised learning. Similar to the philosophy in [6], the objective for ℓ^1 -graph based semi-supervised learning is defined as

$$\min_P \frac{\gamma S_c(P) + (1 - \gamma) \sum_{i=1}^N \left\| P^T x_i - \sum_{j=1}^N W_{ij} P^T x_j \right\|^2}{S_p(P)}$$

where $\gamma \in (0, 1)$ is a threshold for balancing the supervised term and ℓ^1 -graph regularization term, and the supervised part is defined as

$$S_c(P) = \sum_i \sum_{j \in N_{k_1}^+(i)} \|P^T x_i - P^T x_j\|^2 \quad (9)$$

$$S_p(P) = \sum_i \sum_{(i,j) \in P_{k_2}(l_i)} \|P^T x_i - P^T x_j\|^2 \quad (10)$$

where S_c indicates the intraclass compactness, which is represented as the sum of distances between each point and its neighbors of the same class and $N_{k_1}^+(i)$ is the index set of the k_1 nearest neighbors of the sample x_i in the same class, S_p indicates the separability of different classes, which is characterized as the sum of distances between the marginal points and their neighboring points of different classes and $P_{k_2}(l)$ is a set of data pairs that are the k_2 nearest pairs among the set $\{(i, j), l_i = l, l_j \neq l\}$, and W is the weight matrix of the ℓ^1 -graph. Similar to (6), the optimum can be obtained via the generalized eigenvalue decomposition method, and the derived projection matrix P is then used for dimensionality reduction and consequent data classification.

IV. EXPERIMENTS

In this section, we systematically evaluate the effectiveness of ℓ^1 -graph in three learning tasks, namely, data clustering, subspace learning, and semi-supervised learning. For comparison purpose, the classical k -nearest-neighbor graph and ϵ -ball graph with different graph weighting approaches are implemented as evaluation baselines. Note that for all k -near-neighbor graph and ϵ -ball graphs related algorithms, the reported results are based on the tuned best k and ϵ among all proper values.

A. Data Sets

For all the experiments, three databases are used. The USPS handwritten digit database [11] includes ten classes (0–9 digit characters) and 11000 samples in total. We randomly select 200 samples each digit character for the experiments, and all of these images are normalized to the size of 32×32 pixels. The forest covertype database [1] was collected for predicting forest cover type from cartographic variables. It includes seven classes and 581012 samples in total. We randomly select 100 samples for each type in the following experiments. The Extended YALE-B database [13] contains 38 individuals and around 64 near frontal images under different illuminations per individual, where each image is manually cropped and normalized to the size of 32×32 pixels. All the images were taken against a dark homogeneous background with the subjects in an upright and frontal position.

B. Spectral Clustering With ℓ^1 -Graph

In this part of experiments, for a comprehensive evaluation, the ℓ^1 -graph based spectral clustering algorithm is compared with the spectral clustering based on the Gaussian-kernel [17] graph, LE-graphs (used in Laplacian Eigenmaps [5] algorithm), LLE-graphs (ℓ^2 -norm based and used in LLE [16]), and also the K -means clustering results based on the derived low-dimensional representations from principal component analysis

(PCA) [12], and two metrics, the accuracy (AC) and the normalized mutual information (NMI) [21], are used for performance evaluation. Suppose that L is the clustering result label vector and \hat{L} is the known sample label vector, AC is defined as

$$AC = \frac{\sum_{i=1}^N \delta(\hat{L}(i), Map_{(L, \hat{L})}(i))}{N} \quad (11)$$

where N denotes the total number of samples, $\delta(a, b)$ equals to 1 if and only if $a = b$, $Map_{(L, \hat{L})}$ is the best mapping function that permutes X to match Y , where X and Y are the index sets involving all values in L and \hat{L} respectively. The Kuhn–Munkres algorithm is used to obtain the best mapping [7]. On the other hand, the mutual information between X and Y is defined as

$$MI(X, Y) = \sum_{y \in Y} \sum_{x \in X} p(x, y) \log_2 \left(\frac{p(x, y)}{p(x)p(y)} \right) \quad (12)$$

where $p(x)$, $p(y)$ denote the marginal probability distribution functions of X and Y , respectively, and $p(x, y)$ is the joint probability distribution function of X and Y . Suppose $H(X)$ and $H(Y)$ denote the entropies of $p(x)$ and $p(y)$. $MI(X, Y)$ varies between 0 and $\max(H(X), H(Y))$. So, use normalized mutual information NMI as the second metric, namely

$$NMI(X, Y) = \frac{MI(X, Y)}{\max(H(X), H(Y))}. \quad (13)$$

It is obvious that the normalized mutual information NMI takes values in $[0, 1]$. Unlike AC , NMI is invariant with the permutation of labels, namely, NMI does not require the matching X and Y in advance.

The visualization comparison of the data clustering results (digit characters 1–3 from the USPS database) based on ℓ^1 -graph and those based on LE-graph and K -means are depicted in Fig. 3, which shows that the data are much better separated in ℓ^1 -graph. The quantitative comparison results on clustering accuracy are listed in Tables I–III for these three databases respectively. From the listed results, three observations can be made: 1) the clustering results from ℓ^1 -graph based spectral clustering algorithm are consistently much better than those from all other evaluated algorithms for both metrics; 2) (k -nn + LLE)-graph based spectral clustering algorithm is relatively more stable compared with other ones; and 3) ϵ -ball based algorithms show to be generally worse, in both accuracy and robustness, than the corresponding k -nn based graphs, and thus for the consequent experiments, we only report the results from k -nn graphs instead. Note that all the results listed in the tables are from the best tuning of all possible algorithmic parameters, e.g., kernel parameter for G-graph, the number of neighboring samples and ϵ for LE-graphs and LLE-graphs, and the retained feature dimensions for PCA. To further compare the ℓ^1 -norm and ℓ^2 -norm in graph edge weight deduction, we show the clustering accuracies on USPS based on ℓ^1 -graph and (k -nn + LLE)-graphs with variant k in Fig. 4, which shows ℓ^1 -graph is consistently better than ℓ^2 -norm based graph construction for all k 's, and the performance of the latter first increases, and then drops very slowly after k is large enough.

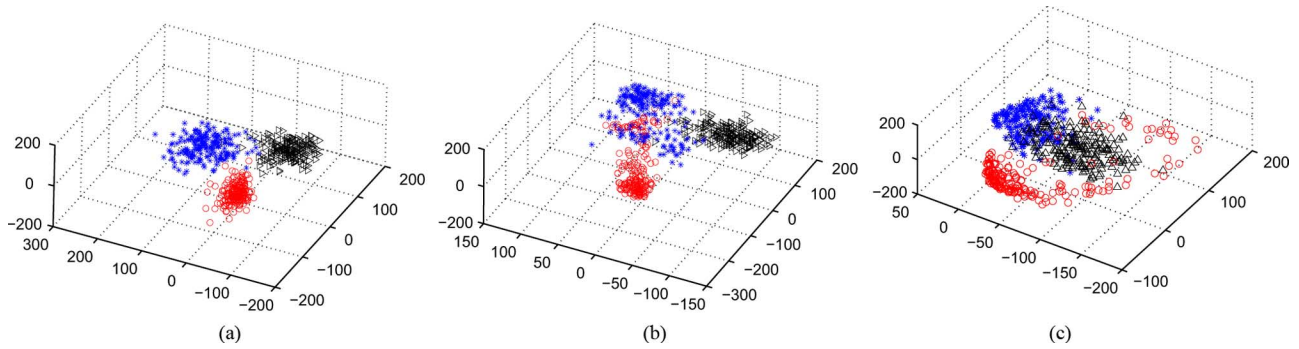


Fig. 3. Visualization of the data clustering results from (a) ℓ^1 -graph, (b) LE-graph, and (c) PCA algorithm for three clusters (handwritten digits 1, 2, and 3 in the USPS database). The coordinates of the points in (a) and (b) are obtained from the eigenvalue decomposition in the third step of Section III-A. Different colors of the points indicate different digits. (a) Clustering from ℓ^1 -graph. (b) Clustering from LE-graph. (c) Clustering via PCA + K-means.

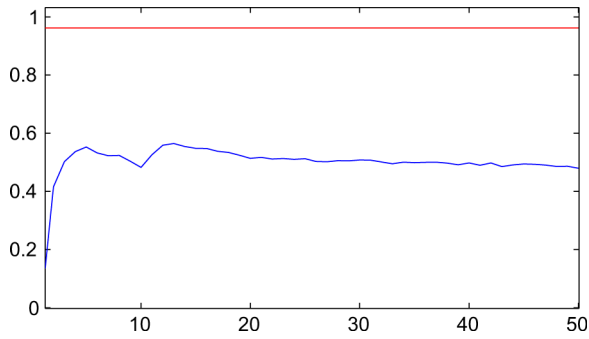


Fig. 4. Comparison clustering accuracies of the ℓ^1 -graph (red line, one fixed value) and $(k\text{-nn} + \text{LLE})$ -graphs (blue curve) with variant k s on the USPS dataset and $K = 7$. It shows that ℓ^1 -norm is superior over ℓ^2 -norm in deducing informative graph weights.

C. Subspace Learning With ℓ^1 -Graph

The experiments on classification based on subspace learning are also conducted on the above three databases. To make the comparison fair, for all the evaluated algorithms we first apply PCA as preprocessing step by retaining 98% energy.

To extensively evaluate the algorithmic performance on the USPS database, we randomly sampled 10, 20, 30, and 40 images from each digit as training data. Similarly, for the forest covertype database, we randomly sampled 5, 10, 15, and 20 samples from each class as training data, and for the Extended YALE-B database, we randomly sampled 10, 20, 30, 40, and 50 training images for each individual. All the remaining data are used for testing purpose. Here, we use the error rate to measure the classification performance, defined as

$$\text{error rate} = 1 - \frac{\sum_{i=1}^{N_t} \delta(\hat{y}_i, y_i)}{N_t} \quad (14)$$

where \hat{y}_i is the predicted sample label and y_i is the given sample label, N_t is the total number of testing samples, and $\delta(\hat{y}_i, y_i)$ equals 0 if $\hat{y}_i \neq y_i$, otherwise equals 1. The best performance of each algorithm over all possible parameters, i.e., graph parameters and feature dimension retained, is reported along with the corresponding feature dimension. The popular unsupervised subspace learning algorithms PCA, NPE, and LPP, and the supervised algorithm Fisherfaces [3] are evaluated for comparison with ℓ^1 -graph based subspace learning, which is essen-

tially unsupervised. For NPE and LPP, we used their unsupervised versions for fair comparison. For LPP, we use the *cosine* metric in graph construction for a better performance. The detailed comparison experimental results for classification are listed in Tables IV–VI for these three databases, from which we can observe: 1) on the forest covertype and Extended YALE-B databases, ℓ^1 -graph based unsupervised subspace learning algorithm generally performs better than the supervised algorithm Fisherfaces, and on the USPS database, Fisherfaces shows a little better than the former; 2) the ℓ^1 -graph based subspace learning algorithm is much superior over all the other evaluated unsupervised subspace learning algorithms; and 3) NPE and LPP show to be better than PCA. Note that for all the classification experiments in this paper, we used the classical nearest neighbor classifier [3], [9], [10] for fairly comparing the discriminating power of the derived subspaces from different subspace learning algorithms. The visualization comparison of the subspaces learnt based on ℓ^1 -graph and those based on PCA, LPP, and NPE are depicted in Fig. 5, from which we can observe bases from PCA show to be most similar to real faces since PCA is motivated for direct data reconstruction.

D. Semi-Supervised Learning With ℓ^1 -Graph

The semi-supervised learning is driven by the philosophy that the unlabeled data can also convey useful information for the learning process. We also use the above three databases for evaluating the effectiveness of the semi-supervised algorithm based on ℓ^1 -graph by comparing with semi-supervised learning algorithms based on Gaussian-kernel graph, LE-graph and LLE-graph. For all the semi-supervised learning algorithms, the supervised part is based on the marginal Fisher analysis [20] algorithm, and the error rate is also used to measure the performances. For a fair comparison, the parameters k_1 , k_2 , and γ are tuned for all proper combinations, and the result reported is based on the best parameter combination. The detailed comparison experiment results for semi-supervised learning algorithms based on different graphs, the original supervised algorithm and the baseline of PCA, are shown in Tables VII–IX, from which we can have two observations: 1) the ℓ^1 -graph based semi-supervised learning algorithm generally achieves the highest classification accuracy compared to semi-supervised learning based on those traditional graphs, and 2) semi-supervised learning can generally bring accuracy

TABLE I

CLUSTERING ACCURACIES (NORMALIZED MUTUAL INFORMATION/NMI AND ACCURACY/AC) FOR SPECTRAL CLUSTERING ALGORITHMS BASED ON ℓ^1 -GRAPH, GAUSSIAN-KERNEL GRAPH (G-GRAPH), LE-GRAPHS, AND LLE-GRAPHS, AS WELL AS PCA + K -MEANS ON THE USPS DIGIT DATABASE. NOTE THAT 1) THE VALUES IN THE PARENTHESES ARE THE BEST ALGORITHMIC PARAMETERS FOR THE CORRESPONDING ALGORITHMS AND FOR THE PARAMETERS FOR AC ARE SET AS THOSE WITH THE BEST RESULTS FOR NMI, AND 2) THE CLUSTER NUMBER K ALSO INDICATES THE CLASS NUMBER USED FOR EXPERIMENTS, THAT IS, WE USE THE FIRST K CLASSES IN THE DATABASE FOR THE CORRESPONDING DATA CLUSTERING EXPERIMENTS

USPS Cluster #	Metric	ℓ^1 -graph	G-graph	LE-graph		LLE-graph		PCA+ K -means
				k -nn	ϵ -ball	k -nn	ϵ -ball	
$K = 2$	NMI	1.000	0.672(110)	0.858(7)	0.627(3)	0.636(5)	0.717(4)	0.608(10)
	AC	1.000	0.922	0.943	0.918	0.917	0.932	0.905
$K = 4$	NMI	0.977	0.498(155)	0.693(16)	0.540(6)	0.606(5)	0.465(7)	0.621(20)
	AC	0.994	0.663	0.853	0.735	0.777	0.668	0.825
$K = 6$	NMI	0.972	0.370(120)	0.682(5)	0.456(6)	0.587(5)	0.427(9)	0.507(4)
	AC	0.991	0.471	0.739	0.594	0.670	0.556	0.626
$K = 8$	NMI	0.945	0.358(150)	0.568(7)	0.371(4)	0.544(12)	0.404(7)	0.462(17)
	AC	0.981	0.423	0.673	0.453	0.598	0.499	0.552
$K = 10$	NMI	0.898	0.346(80)	0.564(6)	0.424(5)	0.552(16)	0.391(4)	0.421(10)
	AC	0.873	0.386	0.578	0.478	0.537	0.439	0.433

TABLE II

CLUSTERING ACCURACIES (NORMALIZED MUTUAL INFORMATION/NMI AND ACCURACY/AC) FOR SPECTRAL CLUSTERING ALGORITHMS BASED ON ℓ^1 -GRAPH, GAUSSIAN-KERNEL GRAPH (G-GRAPH), LE-GRAPHS, AND LLE-GRAPHS, AS WELL AS PCA + K -MEANS ON THE FOREST COVERTYPE DATABASE

COV Cluster #	Metric	ℓ^1 -graph	G-graph	LE-graph		LLE-graph		PCA+ K -means
				k -nn	ϵ -ball	k -nn	ϵ -ball	
$K = 3$	NMI	0.792	0.651(220)	0.554(16)	0.419(6)	0.642(20)	0.475(6)	0.555(5)
	AC	0.903	0.767	0.697	0.611	0.813	0.650	0.707
$K = 4$	NMI	0.706	0.585(145)	0.533(13)	0.534(6)	0.622(20)	0.403(5)	0.522(13)
	AC	0.813	0.680	0.608	0.613	0.782	0.519	0.553
$K = 5$	NMI	0.623	0.561(240)	0.515(12)	0.451(5)	0.556(10)	0.393(7)	0.454(15)
	AC	0.662	0.584	0.541	0.506	0.604	0.448	0.486
$K = 6$	NMI	0.664	0.562(200)	0.545(6)	0.482(6)	0.602(20)	0.465(7)	0.528(8)
	AC	0.693	0.585	0.564	0.523	0.632	0.509	0.547
$K = 7$	NMI	0.763	0.621(130)	0.593(9)	0.452(6)	0.603(11)	0.319(6)	0.602(17)
	AC	0.795	0.642	0.629	0.498	0.634	0.394	0.631

TABLE III

CLUSTERING ACCURACIES (NORMALIZED MUTUAL INFORMATION/NMI AND ACCURACY/AC) FOR SPECTRAL CLUSTERING ALGORITHMS BASED ON ℓ^1 -GRAPH, GAUSSIAN-KERNEL GRAPH (G-GRAPH), LE-GRAPHS, AND LLE-GRAPHS, AS WELL AS PCA + K -MEANS ON THE EXTENDED YALE-B DATABASE. NOTE THAT THE G-GRAPH PERFORMS EXTREMELY BAD IN THIS CASE, A POSSIBLE EXPLANATION OF WHICH IS THAT THE ILLUMINATION DIFFERENCE DOMINATES THE CLUSTERING RESULTS IN G-GRAPH BASED SPECTRAL CLUSTERING ALGORITHM

YALE-B Cluster #	Metric	ℓ^1 -graph	G-graph	LE-graph		LLE-graph		PCA+ K -means
				k -nn	ϵ -ball	k -nn	ϵ -ball	
$K = 10$	NMI	0.738	0.07(220)	0.420(4)	0.354(16)	0.404(3)	0.302(3)	0.255(180)
	AC	0.758	0.175	0.453	0.413	0.450	0.383	0.302
$K = 15$	NMI	0.759	0.08(380)	0.494(4)	0.475(20)	0.438(5)	0.261(5)	0.205(110)
	AC	0.762	0.132	0.464	0.494	0.440	0.257	0.226
$K = 20$	NMI	0.786	0.08(290)	0.492(2)	0.450(18)	0.454(4)	0.269(3)	0.243(110)
	AC	0.793	0.113	0.478	0.445	0.418	0.241	0.238
$K = 30$	NMI	0.803	0.09(50)	0.507(2)	0.417(24)	0.459(7)	0.283(4)	0.194(170)
	AC	0.821	0.088	0.459	0.383	0.410	0.236	0.169
$K = 38$	NMI	0.776	0.11(50)	0.497(2)	0.485(21)	0.473(8)	0.319(4)	0.165(190)
	AC	0.785	0.081	0.443	0.445	0.408	0.248	0.138

TABLE IV

USPS DIGIT RECOGNITION ERROR RATES (%) FOR DIFFERENT SUBSPACE LEARNING ALGORITHMS. NOTE THAT THE NUMBERS IN THE PARENTHESES ARE THE FEATURE DIMENSIONS RETAINED WITH THE BEST ACCURACIES

USPS Train #	Unsupervised				Supervised Fisherfaces
	PCA	NPE	LPP	ℓ^1 -graph-SL	
10	37.21(17)	33.21(33)	30.54(19)	21.91(13)	15.82(9)
20	30.59(26)	27.97(22)	26.12(19)	18.11(13)	13.60(9)
30	26.67(29)	23.46(42)	23.19(26)	16.81(15)	13.59(7)
40	23.25(25)	20.86(18)	19.92(32)	14.35(19)	12.29(7)

TABLE V

FOREST COVER RECOGNITION ERROR RATES (%) FOR DIFFERENT SUBSPACE LEARNING ALGORITHMS

COV Train #	Unsupervised				Supervised Fisherfaces
	PCA	NPE	LPP	ℓ^1 -graph-SL	
5	33.23(17)	28.80(6)	35.09(12)	23.36(6)	23.81(6)
10	27.29(18)	25.56(11)	27.30(16)	19.76(15)	21.17(4)
15	23.75(14)	22.69(16)	23.26(34)	17.85(7)	19.57(6)
20	21.03(29)	20.10(10)	20.75(34)	16.44(6)	18.09(6)

V. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed the concept of ℓ^1 -graph, encoding the overall behavior of the data set in sparse representations. The ℓ^1 -graph is robust to data noises and naturally sparse, and offers adaptive neighborhood for individual datum. It is improvement compared to the counterparts without harnessing extra information from the unlabeled data.

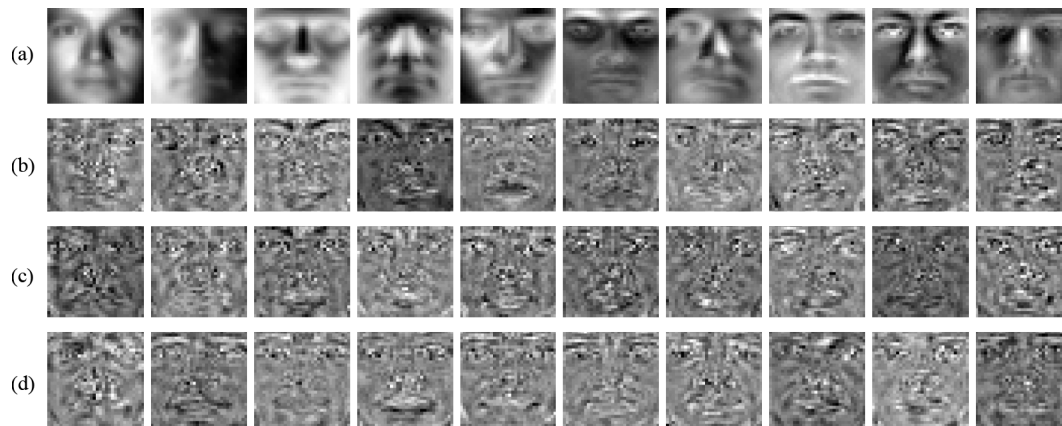


Fig. 5. Visualization comparison of the subspace learning results. They are the first ten basis vectors of (a) PCA, (b) NPE, (c) LPP, and (d) ℓ^1 -graph calculated from the face images in YALE-B database.

TABLE VI
FACE RECOGNITION ERROR RATES (%) FOR DIFFERENT SUBSPACE LEARNING ALGORITHMS ON THE EXTENDED YALE-B DATABASE

YALE-B Train #	Unsupervised				Supervised Fisherfaces
	PCA	NPE	LPP	ℓ^1 -graph-SL	
10	44.41(268)	23.41(419)	24.61(234)	14.26 (112)	13.92 (37)
20	27.17(263)	14.62(317)	14.76(281)	5.30 (118)	9.46(37)
30	20.11(254)	9.40(485)	8.65(246)	3.36 (254)	12.45(34)
40	16.98(200)	5.84(506)	5.30(263)	1.93 (143)	3.79(37)
50	12.68(366)	3.78(488)	3.02(296)	0.75 (275)	1.64(37)

TABLE VII
USPS DIGIT RECOGNITION ERROR RATES (%) FOR DIFFERENT SEMI-SUPERVISED, SUPERVISED, AND UNSUPERVISED LEARNING ALGORITHMS. NOTE THAT THE NUMBERS IN THE PARENTHESES ARE THE FEATURE DIMENSIONS RETAINED WITH THE BEST ACCURACIES

USPS Train #	Semi-supervised			Supervised MFA [20]	Unsupervised PCA
	ℓ^1 -graph	LLE-graph	LE-graph		
10	25.11 (33)	34.63(9)	30.74(33)	34.63(9)	37.21(17)
20	26.94 (41)	41.38(39)	30.39(41)	41.38(39)	30.59(26)
30	23.25 (49)	36.55(49)	27.50(49)	44.34(47)	26.67(29)
40	19.17 (83)	30.28(83)	23.55(83)	35.95(83)	23.35(25)

TABLE VIII
FOREST COVER RECOGNITION ERROR RATES (%) FOR DIFFERENT SEMI-SUPERVISED, SUPERVISED, AND UNSUPERVISED LEARNING ALGORITHMS. NOTE THAT THE NUMBERS IN THE PARENTHESES ARE THE FEATURE DIMENSIONS RETAINED WITH THE BEST ACCURACIES

COV Train #	Semi-supervised			Supervised MFA [20]	Unsupervised PCA
	ℓ^1 -graph	LLE-graph	LE-graph		
5	22.50 (9)	29.89(5)	25.81(7)	29.89(5)	33.23(17)
10	17.45 (10)	24.93(10)	22.74(8)	24.93(10)	27.29(18)
20	15.00 (8)	19.17(10)	17.38(9)	19.17(10)	23.75(14)
30	12.26 (8)	15.32(8)	13.81(10)	16.40(8)	21.03(29)

TABLE IX
FACE RECOGNITION ERROR RATES (%) FOR DIFFERENT SEMI-SUPERVISED, SUPERVISED, AND UNSUPERVISED LEARNING ALGORITHMS ON THE EXTENDED YALE-B DATABASE. NOTE THAT THE NUMBERS IN THE PARENTHESES ARE THE FEATURE DIMENSIONS RETAINED WITH THE BEST ACCURACIES

YALE-B Train #	Semi-supervised			Supervised MFA [20]	Unsupervised PCA
	ℓ^1 -graph	LLE-graph	LE-graph		
5	21.63 (51)	33.47(51)	33.47(51)	33.47(51)	61.34(176)
10	9.56 (61)	18.39(33)	18.39(33)	18.39(33)	44.41(268)
20	5.05 (57)	14.30(29)	11.26(53)	14.30(29)	27.17(263)
30	2.92 (73)	9.15(70)	7.37(71)	11.06(70)	20.11(254)

also empirically observed that the ℓ^1 -graph conveys greater discriminating power compared with classical graphs constructed by k -nearest-neighbor or ϵ -ball method. All these

characteristics make it a better choice for many popular graph-oriented machine learning tasks. We are planning to further study the ℓ^1 -graph from four aspects: 1) beside the learning configurations exploited in this work, there exist many other configurations, e.g., transfer learning, we shall extend the ℓ^1 -graph to these configurations; 2) for multitask (e.g., face recognition, pose estimation and illumination estimation simultaneously) learning problem, an ℓ^1 -graph can only characterize one type of discriminative information, and then how to identify which task the derived graph can contribute to is interesting; 3) currently the ℓ^1 -graph is unsupervised, and how to utilize the label information for constructing a more discriminative graph is another interesting research direction; and 4) currently the computational cost for ℓ^1 -graph construction is relatively high, e.g., about 337 s for the 2414 samples in the YALE-B database on a PC with 3-GHz CPU and 2-GB memory. Then, the entire computational cost is about 364 s for ℓ^1 -graph based subspace learning, while only about 15 s for PCA, 20 s for NPE, and 21 s for LPP. Thus, how to achieve further speedup is very important for large scale applications, e.g., web-scale tasks with millions of images.

REFERENCES

- [1] [Online]. Available: <http://kdd.ics.uci.edu/databases/covertime/covertime.data.html/>
- [2] [Online]. Available: <http://sparselab.stanford.edu>
- [3] P. Belhumeur, J. Hespanha, and D. Kriegeman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 711–720, Jul. 1997.
- [4] M. Belkin, I. Matveeva, and P. Niyogi, "Regularization and semi-supervised learning on large graphs," in *Proc. Int. Conf. Learning Theory*, 2004, vol. 3120, pp. 624–638.
- [5] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Comput.*, vol. 15, no. 6, pp. 1373–1396, 2002.
- [6] D. Cai, X. He, and J. Han, "Semi-supervised discriminant analysis," in *Proc. IEEE Int. Conf. Computer Vision*, 2007, pp. 1–7.
- [7] S. Chen, D. Donoho, and M. Saunders, "Atomic decomposition by basis pursuit," *Soc. Ind. Appl. Math. Rev.*, vol. 43, no. 1, pp. 129–159, 2001.
- [8] D. Donoho, "For most large underdetermined systems of linear equations the minimal ℓ^1 -norm solution is also the sparsest solution," *Commun. Pure Appl. Math.*, vol. 59, no. 7, pp. 797–829, 2004.
- [9] X. He, D. Cai, S. Yan, and H. Zhang, "Neighborhood preserving embedding," in *Proc. IEEE Int. Conf. Computer Vision*, 2005, vol. 2, pp. 1208–1213.
- [10] X. He and P. Niyogi, "Locality preserving projections," *Adv. Neural Inf. Process. Syst.*, vol. 16, pp. 585–591, 2003.

- [11] J. Hull, "A database for handwritten text recognition research," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 5, pp. 550–554, May 1994.
- [12] I. Jolliffe, *Principal Component Analysis*. New York: Springer-Verlag, 1986, pp. 1580–1584.
- [13] K. Lee, J. Ho, and D. Kriegman, "Acquiring linear subspaces for face recognition under variable lighting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 5, pp. 684–698, May 2005.
- [14] N. Meinshausen and P. Bühlmann, "High-dimensional graphs and variable selection with the lasso," *Ann. Statist.*, vol. 34, no. 3, pp. 1436–1462, 2006.
- [15] B. Olshausen and D. Field, "Sparse coding with an overcomplete basis set: A strategy employed by v1?," *Vis. Res.*, vol. 37, no. 23, pp. 3311–3325, 1998.
- [16] S. Roweis and L. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.
- [17] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.
- [18] J. Tenenbaum, V. Silva, and J. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.
- [19] J. Wright, A. Ganesh, A. Yang, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, Feb. 2009.
- [20] S. Yan, D. Xu, B. Zhang, Q. Yang, H. Zhang, and S. Lin, "Graph embedding and extensions: A general framework for dimensionality reduction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 1, pp. 40–51, Jan. 2007.
- [21] X. Zheng, D. Cai, X. He, W. Ma, and X. Lin, "Locality preserving clustering for image database," in *Proc. ACM Int. Conf. Multimedia*, 2004, pp. 885–891.
- [22] X. Zhu, Z. Ghahramani, and J. Lafferty, "Semi-supervised learning using gaussian fields and harmonic functions," in *Proc. Int. Conf. Machine Learning*, 2003, pp. 912–919.

Bin Cheng received the B.E. degree from the Department of Electronics Engineering and Information Science, University of Science and Technology of China (USTC), China, in 2007. He is currently pursuing the Ph.D. degree at the National University of Singapore (NUS).

Since Fall 2008, he has been with the Department of Electrical and Computer Engineering, NUS. He is currently working with Prof. S. Yan on his Ph.D. degree. His research interests include image processing, computer vision, and machine learning.

Jianchao Yang (S'08) received the B.E. degree from the Department of Electronics Engineering and Information Science, University of Science and Technology of China (USTC), China, in 2006. He is currently pursuing the Ph.D. degree under Prof. T. S. Huang.

Since Fall 2006, he has been with the Department of Electrical and Computer Engineering of University of Illinois at Urbana-Champaign (UIUC), Urbana. His research interests include image processing, computer vision, and machine learning.

Shuicheng Yan (M'06–SM'09) received the Ph.D. degree from the School of Mathematical Sciences, Peking University, in 2004.

He spent three years as Postdoctoral Fellow at the Chinese University of Hong Kong and then at the University of Illinois at Urbana-Champaign, Urbana, and he is currently an Assistant Professor in the Department of Electrical and Computer Engineering at the National University of Singapore. In recent years, his research interests have focused on computer vision (biometrics, surveillance, and internet vision), multimedia (video event analysis, image annotation, and media search), machine learning (feature extraction, sparsity/non-negativity analysis, large-scale machine learning), and medical image analysis. He has authored or coauthored over 140 technical papers over a wide range of research topics.

Dr. Yan has served on the editorial board of the *International Journal of Computer Mathematics*, as guest editor of a special issue of *Pattern Recognition Letters*, and as a guest editor of a special issue of *Computer Vision and Image Understanding*. He has served as Co-Chair of the IEEE International Workshop on Video-oriented Object and Event Classification (VOEC'09) held in conjunction with ICCV'09. He is the special session chair of the Pacific-Rim Symposium on Image and Video Technology 2010. He is an Associate Editor of IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY.

Yun Fu (S'07–M'08) received the B.Eng. degree in information engineering from the School of Electronic and Information Engineering, Xi'an Jiaotong University, China, in 2001; the M.Eng. degree in pattern recognition and intelligence systems from the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, in 2004; the M.S. degree in statistics from the Department of Statistics, University of Illinois at Urbana-Champaign (UIUC), Urbana, in 2007; and the Ph.D. degree in electrical and computer engineering from the Electrical and Computer Engineering (ECE) Department, UIUC, in 2008.

From 2001 to 2004, he was a research assistant at the Institute of Artificial Intelligence and Robotics at XJTU. From 2004 to 2008, he was a graduate fellow and research assistant at the Beckman Institute for Advanced Science and Technology, ECE Department and Coordinated Science Laboratory, UIUC. He was a research intern with Mitsubishi Electric Research Laboratories, Cambridge, MA, Summer 2005 and with the Multimedia Research Lab of Motorola Labs, Schaumburg, IL, Summer 2006. He joined BBN Technologies, Cambridge, MA, as a Scientist in 2008 to build and lead the computer vision and machine learning team. He held a part-time Lecturer position at the Computer Science Department, Tufts University, Medford, MA, in the spring of 2009. He joined the Computer Science and Engineering Department, University at Buffalo, The State University of New York, as an Assistant Professor in 2010. His research interests include machine learning, human computer interaction, image processing, multimedia, and computer vision. He has extensive publications in top journals, book chapters, and international conferences/workshops.

Dr. Fu has served as associate editor, chair, PC member, and reviewer for many top journals and international conferences/workshops. He is the recipient of the 2002 Rockwell Automation Master of Science Award, two Edison Cups of the 2002 GE Fund Edison Cup Technology Innovation Competition, the 2003 HP Silver Medal and Science Scholarship, the 2007 Chinese Government Award for Outstanding Self-financed Students Abroad, the 2007 DoCoMo USA Labs Innovative Paper Award (IEEE ICIP'07 best paper award), the 2007–2008 Beckman Graduate Fellowship, and the 2008 M. E. Van Valkenburg Graduate Research Award. He is a life member of Institute of Mathematical Statistics (IMS) and Beckman Graduate Fellow.



Thomas S. Huang (LF'01) received the B.S. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, R.O.C., and the M.S. and D.Sc. degrees in electrical engineering from the Massachusetts Institute of Technology (MIT), Cambridge.

He was on the Faculty of the Department of Electrical Engineering, MIT, from 1963 to 1973, and on the Faculty of the School of Electrical Engineering and Director of its Laboratory for Information and Signal Processing at Purdue University, West Lafayette, IN, from 1973 to 1980. In 1980, he joined the University of Illinois at Urbana-Champaign, where he is now the William L. Everitt Distinguished Professor of Electrical and Computer Engineering and Research Professor at the Coordinated Science Laboratory and Head of the Image Formation and Processing Group at the Beckman Institute for Advanced Science and Technology and Co-Chair of the Institute's major research theme Human Computer Intelligent Interaction. His professional interests lie in the broad area of information technology, especially the transmission and processing of multidimensional signals. He has published 20 books and over 500 papers in network theory, digital filtering, image processing, and computer vision.

Dr. Huang is a Member of the National Academy of Engineering; a Foreign Member of the Chinese Academies of Engineering and Sciences; and a Fellow of the International Association of Pattern Recognition and the Optical Society of American. He has received a Guggenheim Fellowship, an A.V. Humboldt Foundation Senior U.S. Scientist Award, and a Fellowship from the Japan Association for the Promotion of Science. He received the IEEE Signal Processing Society's Technical Achievement Award in 1987 and the Society Award in 1991. He was awarded the IEEE Third Millennium Medal in 2000. Also in 2000, he received the Honda Lifetime Achievement Award for "contributions to motion analysis". In 2001, he received the IEEE Jack S. Kilby Medal. In 2002, he received the King-Sun Fu Prize, International Association of Pattern Recognition, and the Pan Wen-Yuan Outstanding Research Award. In 2005, he received the Okawa Prize. In 2006, he was named by IS&T and SPIE as the Electronic Imaging Scientist of the year. He is a Founding Editor of the *International Journal Computer Vision, Graphics, and Image Processing* and Editor of the Springer Series in Information Sciences, published by Springer Verlag.