A Unified Feature Selection Framework for Graph Embedding on High Dimensional Data

Marcus Chen, Ivor W. Tsang, Mingkui Tan, and Tat Jen Cham

Abstract—Although graph embedding has been a powerful tool for modeling data intrinsic structures, simply employing all features for data structure discovery may result in noise amplification. This is particularly severe for high dimensional data with small samples. To meet this challenge, this paper proposes a novel efficient framework to perform feature selection for graph embedding, in which a category of graph embedding methods is cast as a least squares regression problem. In this framework, a binary feature selector is introduced to naturally handle the feature cardinality in the least squares formulation. The resultant integral programming problem is then relaxed into a convex Quadratically Constrained Quadratic Program (QCQP) learning problem, which can be efficiently solved via a sequence of accelerated proximal gradient (APG) methods. Since each APG optimization is w.r.t. only a subset of features, the proposed method is fast and memory efficient. The proposed framework is applied to several graph embedding learning problems, including supervised, unsupervised, and semi-supervised graph embedding. Experimental results on several high dimensional data demonstrated that the proposed method outperformed the considered state-of-the-art methods.

Index Terms—Sparse graph embedding, sparse principal component analysis, efficient feature selection, high dimensional data

1 Introduction

High dimensional data is ubiquitous in many real world applications, especially in this era of data [1], [2]. In microarray technology, a large sensor array can capture thousands of genes simultaneously. The latest camera phones such as the Nokia Lumia 1020 are able to capture photos of up to 41 megapixels. However, directly learning a classifier on high dimensional data may significantly degrade the performance of many applications, especially when data features are highly correlated and the sample size is relatively small. This is commonly referred to as *the curse of dimensionality* [3]. To alleviate this, one possible approach is to transform high dimensional data into a lower dimensional representation while preserving the intrinsic data structure. This is known as dimensionality reduction.

Graph embedding has been shown to be a powerful tool for dimensionality reduction [4], [5]. In particular, some popular dimensionality reduction methods such as Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA) [6], Isomap [7], Locally Linear Embedding (LLE) [8], and Locality Preserving Projection (LPP) [4] can be formulated into graph embedding methods [5]. By employing full dimensional features for learning tasks, the graph embedding methods aim to learn a low dimensional projection, preserving some intrinsic data structures. Intrinsic data structures can have both local and global properties, depending on the applications. Local properties often refer to the local neighborhood relationship such as in LPP, while examples of global properties include class separation in LDA, the global variance in PCA, and the global shortest path between any pairs of data samples in the Isomap method. Graph embedding of high dimensional data suffers from two weaknesses. First, it is hard to interpret the resultant features when using all dimensions for embedding. Second, the original data inevitably contains noisy feature measurements. Simply incorporating these noisy features could make graph embedding unreliable and noisy [9], [10]. Therefore, it is important to select only the significant features for graph embedding.

Many feature selection methods have been proposed in different learning contexts [11], [12], [13], [14], [15], [16] with different feature importance measures. These methods can be categorized into two classes, namely, the supervised and unsupervised methods [17]. For the supervised methods, there are two main feature importance measures, distance based measures, and the correlation based measures. Specifically, the distance based measures define the important features as those that separate classes better and cluster the within-class samples, such as LDA-based feature selection methods [18]. In correlation based feature selection methds [19], the important features are those that correlate well with class labels and give better prediction results. In the unsupervised methods, due to the absence of class labels, several criteria have been proposed to evaluate the feature importance based on different learning contexts, such as information measure, variance measure, and locality measure, summarized as follows:

- *M. Chen and T. J. Cham are with the School of Computer Engineering, Nanyang Technological University, Singapore. E-mail: marcuschen@pmail.ntu.edu.sg, astjcham@ntu.edu.sg.*
- *I. W. Tsang is with Centre for Quantum Computation and Intelligent Systems, University of Technology Sydney, Sydney, NSW, Australia. E-mail: ivor.tsang@gmail.com.*
- *M. Tan is with the School of Computer Science, The University of Adelaide, Adelaide, SA, Australia. E-mail: tanmingkui@gmail.com.*

Manuscript received 9 Jan. 2014; revised 8 Oct. 2014; accepted 24 Nov. 2014. Date of publication 17 Dec. 2014; date of current version 27 Apr. 2015. Recommended for acceptance by T. Li.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TKDE.2014.2382599

- Information measure: good features gain more information when included.
- Variance measure: good features capture more variance in the data.
- Locality measure: good features can preserve data locality better.

The existing feature selection methods are also often classified as filter, wrapper, and embedded approaches [20]. Filter methods aim to predefine some desired intrinsic data properties and rank feature subsets accordingly. These methods employ either forward selection or backward elimination search strategies, often resulting in local optimality. They do not consider the subsequent applications for feature evaluation, and are thus suboptimal in performance. He et al. [21] proposed to encode data locality via a Laplacian Score (LS) for feature ranking. Similarly, Zhao and Liu [22] proposed to evaluate features based on SPECtrum decomposition (SPEC) on the Laplacian matrix. These two methods employ feature level ranking, and do not consider the redundancy among features. To address this, Nie et al. [23] proposed a feature subset trace ratio method to rank feature subsets that best discriminate between-class samples and preserve within-class relationships.

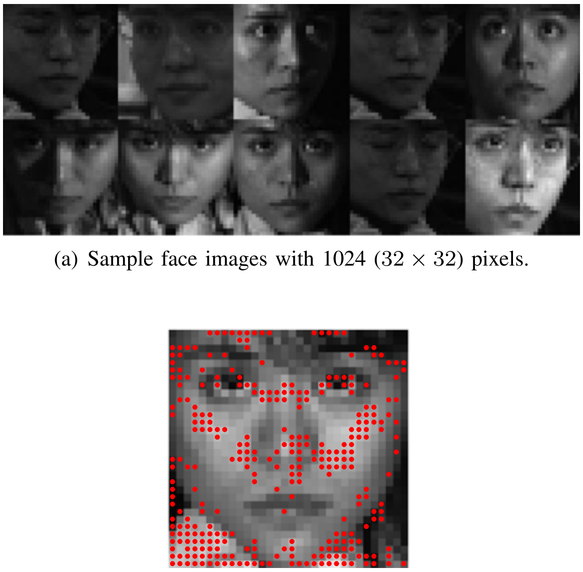
In contrast to filter methods, wrapper methods are application dependent. The feature subset giving the best performance of a predetermined learning task will be chosen. The evaluation of each feature subset requires a complete run of the application, and is thus very computationally expensive. Generally, given a learning task, wrapper methods can perform better than filter methods [24]. Finally, embedded methods encapsulate the feature selection problem into their optimization objectives such as the solution sparsity [25], [26], [27], [28].

Among the sparsity induced methods, the sparse regression method, termed as LASSO [29], is popular in the literature due to its simplicity and efficiency in computation. Its formulation of the feature selection problem is as follows:

$$\min_{w} \quad \|X^{\top} w - y\|^2 + \gamma \|w\|_1, \tag{1}$$

where $X \in \mathbb{R}^{d \times n}$ is the d-dimensional data matrix of n samples, y is the response variable vector, and w is the weight vector. The regularization constant γ controls the sparsity of solution; a larger γ generally results in a sparser solution, and vice versa. However, LASSO can only choose at most n variables even when $d \geq n$, and the solution is not well defined unless $\|w\|_1$ is bounded by a certain value [30]. The LASSO shrinkage may produce biased estimates for large coefficients of w [31]. Zhou and Hastie [30] then proposed the Elastic net method, which adds an L_2 norm term, i.e., $\gamma\|w\|_1 + \beta\|w\|^2$. However, both LASSO and the Elastic net method do not guarantee the choice of the same set of features when regressing over different principal components (y). To address this issue, Cai et al. [32] proposed a simple heuristic method to rank the features based on the weight of different subspaces. This method is termed as MultiClusters Feature Selection (MCFS) with preserving data locality as its main learning objective.

Recently, the Convex Relaxations for Subset Selection method (CRSS) [33] was shown to achieve better results than the Least Angle Regression (LAR) method [34]. Bach et al. [33] proposes two search methods. One is a branch and bound search method and is combinatorial in computational complexity. The other one is based on the random generation of Gaussian vectors. Both LASSO and CRSS select features based on sequential iterations of principal components. In this manner, tuning parameters for different components is complicated [10]. Therefore, they do not optimize for the collective objectives. There has been little work of estimating multiple sparse components simultaneously. The most similar method by Vu et al. [10], Fantope Projection and Selection (FPS), optimizes the collective objectives for sparse PCA via a Fantope projection approach.


(a) Sample face images with 1024 (32 × 32) pixels.

(b) Overlaying a face sample with 300 features using the proposed method.

Fig. 1. Important pixels that capture variations in the face images under different illumination conditions, poses, and facial expressions.

Both the graph embedding and feature selection methods define their own paradigms to preserve data intrinsic structures. In existing work in the literature, these two tasks have been done independently or mutually exclusively. This paper instead proposes a novel paradigm to unify these two schemes by performing feature selection and graph embedding simultaneously. As an application illustrated in Fig. 1b, the proposed paradigm is applied to the LPP graph embedding of some face images. The proposed method can automatically choose some important pixels that capture the variations in illumination conditions, poses, and facial expressions. The main contributions of this paper are summarized as follows.

- By exploiting the least squares formulation of graph embedding, we introduce a binary feature selector to directly constrain the desired number of features. We further reformulate the resultant problem as a convex semi-infinite programming problem (SIP). This novel feature selection scheme can be applied to unsupervised, supervised, and semi-supervised learning tasks in preserving the corresponding intrinsic data structures via low dimensional embeddings.
- By exploiting the observation that only a few constraints are active in the resultant SIP problem, we

proposed an efficient cutting plane method, which essentially conducts a sequence of accelerated proximal gradients on a set of features only. Therefore, a major advantage of the proposed method is its ability to handle ultrahigh dimensions efficiently due to its low computation cost and memory requirements. Moreover, the proposed method is guaranteed to converge globally.

- The proposed method addresses the learning in a holistic way, resulting in both generalized graph embedding and the desired cardinality of the features. A wide range of datasets have been tested in the experiments to verify the effectiveness of the proposed framework for unsupervised, supervised, and semi-supervised learning tasks.

The organization of the rest of the paper is as follows. In Section 2, we briefly review some graph embedding methods for dimensionality reduction and introduce the least squares formulation of graph embedding. Section 3 details the proposed approach. In Section 4, we conduct some experiments to compare our results with the current state-of-the-art algorithms. Section 5 concludes this paper.

2 Related Work and Preliminaries

In this section, we first briefly review the recent literature on graph embedding and feature selection. An overview of the generalized subspace problem is also provided since this is a foundation of our method.

2.1 Graph Embedding for Dimensionality Reduction

Coherent structures in high dimensional data, such as neighboring pixels in images, naturally induce a high correlation among dimensions. To alleviate the curse of dimensionality, scientists have proposed to transform data into a low dimensional manifold via graph embedding [5], [21]. The number of data samples, data dimensionality, and the number of classes are denoted by n, d, and k, respectively. $X \in \mathbb{R}^{d\times n}$ is a zero-mean data matrix, and $Y = \{y_1, y_2, \dots y_n\} \in \mathbb{R}^{k\times n}$ if available represents class label information. Furthermore, the symmetric positive semi-definite matrix, $S \in \mathbb{R}^{n\times n}$, encodes the desired data properties. Graph embedding for a class of dimensionality reduction methods aims to find the projection vector ω for the following generalized eigenvalue problem [35]:

$$XSX^\top \omega = \lambda XX^\top \omega. \tag{2}$$

Many dimensionality reduction methods such as PCA, LDA, CCA, LPP, and Hypergraph Spectral Learning (HSL) can be formulated into the above graph embedding framework [5]. The definitions of S for the above mentioned methods are tabulated in Table I. More details can be found in [35] and references therein.

TABLE 1
S Computations for Common Generalized Eigen Problems

S/N	Methods	S
1	**PCA**	$X^\top X$
2	**CCA**	$Y^\top(YY^\top)^{-1}Y$
3	**LDA**	$Y^\top(YY^\top)^{-1}Y$
4	**LPP**	$D^{-1/2}AD^{-1/2}$
5	**HSL**	$I - L$
6	**SDA**	$(I+\beta D)^{-1/2}(W_b+\beta A)(I+\beta D)^{-1/2}$

X is a data matrix, Y is a regression response matrix in CCA and class label matrix in LDA and PLS, A is an affinity matrix, D is a diagonal matrix whose diagonal entries are the row sum of A, L is a Laplacian matrix, and β is a regularization constant.

Both PCA and LPP are unsupervised as they do not consider class labels. They are often used for general pre-processing, clustering, or visualization. For classification, supervised graph embedding, such as LDA, generally can achieve better performance since class label information is considered. The graph embedding can be readily extended to the semi-supervised setting, which utilizes large unlabeled datasets and small labeled datasets to model intrinsic data structures [36], [37], [38], [39]. To achieve this, a possible model is a weighted graph whose vertices are both labeled and unlabeled samples and edges reflect samples similarity. For example, the semi-supervised discriminative analysis (SDA) method [40] builds upon the LDA and LPP graph embeddings.

2.2 The Least Squares Formulation for Graph Embedding

Solving the generalized eigenvalue problem in (2) is very expensive for large-scale and high dimensional problems. To reduce the computational burden for large-scale problems, Sun et al. [35] formulates it into a least squares problem, as in (4). Specifically, since $S \in \mathbb{R}^{n\times n}$ is a symmetric positive semi-definite matrix, it can be decomposed as $S = HH^\top$, where $H \in \mathbb{R}^{n\times r}$ and $r \leq n$ is the number of significant singular values of S. Furthermore, let $HP = QR$ be the *QR-decomposition* of H with the permutation matrix P, where $Q \in \mathbb{R}^{n\times r}$ is a matrix with r orthonormal columns. Moreover, let $R = U_R\Sigma_R V_R^\top$ be the compact singular value decomposition (SVD) of R, and the regression response $T \in \mathbb{R}^{n\times r}$ is computed as follows:

$$T = QU_R. \tag{3}$$

Then the generalized eigenvalue problem can be cast as the following least squares regression problem,

$$\min_W \quad \|X^\top W - T\|_F^2, \tag{4}$$

where $X \in \mathbb{R}^{d\times n}$ and $W \in \mathbb{R}^{d\times r}$ are data matrix and weight matrix, respectively. In practice, to improve the robustness to noise and avoid overfitting, a regularization term could be added as follows:

$$\min_W \quad \|X^\top W - T\|_F^2 + \gamma\|W\|_F^2. \tag{5}$$

3 General Framework For Feature Selection

After transforming the generalized eigenvalue problem into a regression problem, the formulation (5) can benefit from many existing efficient least squares solvers. However, the regularizer $\gamma\|W\|_F^2$ may not produce sparse solutions. In other words, it cannot achieve the feature selection task. To induce sparsity, we introduce a binary vector $p \in \{0,1\}^d$, whose entries are 1 for the selected features and 0 otherwise. To select m desired features, exactly m entries in p will be set to 1, where $m \ll d$. Let $\mathbb{P} = \{p : p \in \{0,1\}^d, p^\top \mathbf{1} = m\}$ be the domain of p, and $\mathbf{1} \in \mathbb{R}^d$ denotes a vector with all entries equal to 1. The proposed least squares formulation for graph embedding based feature selection is cast as the following optimization problem,

$$\begin{aligned} \min_{p \in \mathbb{P}} \min_{W} \quad & \frac{1}{2}\|\Xi\|_F^2 + \frac{\gamma}{2}\|W\|_F^2 \\ s.t. \quad & \Xi = X^\top \mathrm{diag}(p) W - T, \end{aligned} \tag{6}$$

where $T \in \mathbb{R}^{n\times r}$ is the response matrix, $\mathrm{diag}(p)$ is the matrix whose diagonal is the feature selector vector p, and $\Xi \in \mathbb{R}^{n\times r}$ denotes the residual matrix.

The proposed formulation has several advantages over the conventional approaches, which impose sparsity directly on W like the sparsity objective $\sum_i \|W_i\|_1$ in the sparse PCA (SPCA) method [41]. First, it selects features naturally with the desired cardinality. This is much more efficient than the sparsity induced methods, in which a regularizer constant controls cardinality. Second, the proposed model can be transformed to a convex programming problem [42], based on which an efficient solver can be developed. The similar schemes used in [42] and [18] are designed for the Fisher score method and classification method, respectively. These two methods can be seen as special cases of the proposed framework in (6).

In general, the problem in (6) is NP-hard to solve due to the combinatorial integral constraints on p. To address it, it is necessary to make some transformations and relaxations. It is not difficult to verify that the inner minimization problem with a fixed p can be solved equivalently in its dual. By introducing $V \in \mathbb{R}^{n\times r}$, the dual variable, to the constraint $\Xi = X^\top \mathrm{diag}(p) W - T$, we can solve the inner regression problem in its dual. Specifically, the Lagrangian function of the inner regression problem is

$$\mathcal{L}(W, \Xi, V) = \frac{1}{2}\|\Xi\|_F^2 + \frac{\gamma}{2}\|W\|_F^2 + \langle V, \Xi - X^\top \mathrm{diag}(p) W + T\rangle,$$

where $\langle \cdot,\cdot\rangle$ denotes the inner product. By setting the first derivatives of $\mathcal{L}(W,\Xi,V)$ w.r.t. W and Ξ to zero, we can obtain the Karush-Kuhn-Tucker (KKT) conditions, namely, $\gamma W = \mathrm{diag}(p) X V$ and $V = -\Xi$. By substituting these results into the Lagrangian function, the problem in (6) can be transformed into the following dual formulation:

$$\min_{p\in\mathbb{P}} \max_{V\in\mathbb{R}^{n\times k}} f(V,p), \tag{7}$$

where

$$f(V,p) = tr(V^\top T) - \frac{1}{2} tr\left(V^\top \left(\frac{1}{\gamma} X^\top \mathrm{diag}(p) X + I\right) V\right).$$

However, this problem is still a non-convex problem since the main optimization variable p is in discrete values. Following the convex relaxation in [42], we have

$$\min_{p\in\mathbb{P}} \max_{V\in\mathbb{R}^{n\times k}} f(V,p) \geq \max_{V\in\mathbb{R}^{n\times k}} \min_{p\in\mathbb{P}} f(V,p).$$

Moreover, this convex relaxed problem can be further transformed into a convex QCQP problem by introducing an additional variable $\theta \in \mathbb{R}$,

$$\begin{aligned} \max_{V\in\mathbb{R}^{n\times k}, \theta\in\mathbb{R}} \quad & \theta \\ s.t. \quad & \theta \leq f(V,p), \;\; p \in \mathbb{P}. \end{aligned} \tag{8}$$

Note that the constraint domain $\mathbb{P}$ contains a combinatorial number of p's, making the optimization problem intractable even for small-sized p and m.

3.1 Sparse Graph Embedding for Feature Selection

The optimization problem in (8) has a combinatorial number of constraints. However, only a few of them are active. Exploiting this observation, we adopt the cutting plane algorithm to solve the QCQP problem (8). The cutting plane algorithm iteratively finds the most active constraint and adds it to the active constraint set Π, which is initialized to an empty set $\emptyset$. The active constraint set Π is always a subset of $\mathbb{P}$, i.e., $\Pi \subseteq \mathbb{P}$. Given the updated set Π, we solve the following subproblem,

$$\begin{aligned} \max_{V\in\mathbb{R}^{n\times k}, \theta\in\mathbb{R}} \quad & \theta \\ s.t. \quad & \theta \leq f(V,p^t), \;\; \forall p^t \in \Pi. \end{aligned} \tag{9}$$

We term our proposed procedure Sparse Graph Embedding (or **SparGE** for short), described in Algorithm 1.

Algorithm 1. Sparse Graph Embedding for Feature Selection

```
Input: data $X \in \mathbb{R}^{d\times n}$, a positive semi-definite matrix $S$, the desired feature cardinality $m$.
  1) Initialize $\Pi = \emptyset$, and compute $T$ according to (3). Assign $t := 1$.
  2) Iterate the following two steps until convergence.
      a) Update $V$ by solving the subproblem in (9).
      b) Find the most active constraint, which is indicated by $p^t$, by solving $p^t = \mathrm{argmax}_p f(V,p)$, based on $V$.
         Update $\Pi$ by $\Pi := \Pi \cup \{p^t\}$ and $t$ by $t := t + 1$.
Output: $\Pi = \{p^1, p^2, \ldots, p^k\}$, with each $p^i$ indexing the selected features.
```

Given V, Step 2b) of Algorithm 1 requires us solving

$$p^t = \underset{p}{\mathrm{argmax}}\, f(V,p) = \underset{p}{\mathrm{argmax}}\, \|\mathrm{diag}(p) X V\|_F^2$$

in order to find the most active constraint of problem (8). Let $A = XV \in \mathbb{R}^{d\times r}$, and define $s_i = \sum_{j=1}^r (A_{i,j})^2$. The optimization problem becomes:

$$\underset{p}{\mathrm{argmax}}\, \|\mathrm{diag}(p) X V\|_F^2 = \underset{p}{\mathrm{argmax}} \sum_{i=1}^d s_i p_i. \tag{10}$$

Apparently, problem (10) can be solved readily by sorting s and then setting its top m values in s to 1 and the rest to 0. In other words, the most active constraint can be identified by choosing the features with the m highest values in s. The algorithm for the most active constraint analysis is summarized in Algorithm 2. The most active constraint p^t obtained is then added to the active constraint set $\Pi := \Pi \cup \{p^t\}$.

Algorithm 2. The Most Active Constraint Selection

```
Input: data $X \in \mathbb{R}^{d\times n}$, dual variable $V$, the desired number of features $m$, and the selection vector $p$.
  1) Set all the entries of $p$ to 0.
  2) Compute $s_i = \sum_{j=1}^{k}(A_{i,j})^2, \forall i = 1,\ldots,d$.
  3) Sort $s$ in descending order.
  4) Set $m$ entries of $p$ w.r.t. the top $m$ values of $s$.
Output: $p$ which defines the most active constraint.
```

3.2 The Subproblem Optimization

After updating the active constraint set Π, we then solve the subproblem in (9) with reduced constraints as defined by Π. Since the number of constraints in Π is no longer large, this problem is readily solved by a sub-gradient method, such as simpleMKL [18], [42]. However, solving this problem w.r.t. the dual variables V can be very expensive, in particular when n is very large.

Assume there are κ active constraints in Π, i.e., $\kappa = |\Pi|$. Even though there are a large number of features in X, at most $m\kappa$ features are chosen by $\Pi \subseteq \mathbb{P}$, where $m\kappa \ll d$. Based on this observation, the subproblem in (9) might be solved more efficiently w.r.t. the primal variables W. To be more specific, following [43], we have the following proposition.

Proposition 1. *The subproblem in (9) can be equivalently addressed in the following primal form:*

$$\min_{\Omega} \frac{\gamma}{2}\left(\sum_{t=1}^{C}\|\Omega^t\|_F\right)^2 + \frac{1}{2}\|\Xi\|_F^2, \tag{11}$$

where $\Xi = T - \sum_{t=1}^{\kappa} X^\top \mathrm{diag}(p^t)\Omega^t$ denotes the regression residual matrix and Ω^t denotes the weight matrix defined on the features indicated by p^t. Moreover, the dual variable V in (9) can be recovered by $V = \Xi$, which is required for the most active constraint selection.

The proof of this proposition is included in Appendix A. Problem (11) is a non-smooth problem due to the regularization term $\frac{\gamma}{2}(\sum_{t=1}^{C}\|\Omega^t\|_F)^2$. However, there are at most $m\kappa$ (where $m\kappa \ll d$) features involved in this problem, making it easier to be solved.[1] For convenience, we define $\Omega = [\Omega^1, \Omega^2, \ldots, \Omega^\kappa] \in \mathbb{R}^{d\times\kappa r}$ by stacking $\Omega^i \in \mathbb{R}^{d\times r}$. Let $P(\Omega) = \frac{\gamma}{2}(\sum_{t=1}^{\kappa}\|\Omega^t\|_F)^2$ and $f(\Omega) = \frac{1}{2}\|\Xi\|_F^2$. Following [44], we propose to solve the primal problem using the accelerated proximal gradient method (APG), which iteratively minimizes the following quadratic approximation of (11):

$$\begin{aligned} Q(\Omega, \Omega_t) &= f(\Omega_t) + < \nabla f, \Omega - \Omega_t > + \frac{\tau}{2}\|\Omega - \Omega_t\|_F^2 + P(\Omega) \\ &= \frac{\tau}{2}\|\Omega - G\|_F^2 + P(\Omega) + f(\Omega_t) - \frac{1}{2\tau}\|\nabla f\|_F^2, \end{aligned} \tag{12}$$

where ∇f denotes the gradient of f at point Ω_t, $\tau > 0$ denotes the Lipschitz constant of $f(\Omega)$, and $G = \Omega_t - \frac{1}{\tau}\nabla f = [G^1, G^2, \ldots, G^\kappa] \in \mathbb{R}^{d\times\kappa r}$ w.r.t. $\Omega = [\Omega^1, \Omega^2, \ldots, \Omega^\kappa]$. Note that $f(\Omega_t) - \frac{1}{2\tau}\|\nabla f\|_F^2$ is constant w.r.t. Ω, and thus we just need to solve the following projection problem:

$$\min_{\Omega} \frac{\tau}{2}\|\Omega - G\|_F^2 + P(\Omega). \tag{13}$$

This problem has a unique global closed-form solution, which can be calculated as follows via Moreau Projection [45].

Proposition 2. *Suppose the optimal solution to problem (13) is $S_\tau(G) = [S_\tau(G^1), S_\tau(G^2), \ldots, S_\tau(G^\kappa)] \in \mathbb{R}^{d\times\kappa r}$ and $o = [o_1, o_2, \ldots, o_\kappa]' \in \mathbb{R}^\kappa$ is an intermediate variable. Then $S_\tau(G)$ is unique and its tth component, $S_\tau(G^t)$, can be calculated as follows:*

$$S_\tau(G^t) = \begin{cases} \frac{o_t}{\|G^t\|_F}G^t, & if\ \ o_t > 0, \\ \mathbf{0}, & otherwise. \end{cases} \tag{14}$$

where $t \in \{1, 2, \ldots, \kappa\}$. The intermediate vector o_t can be calculated via a soft-threshold operator $\mathrm{soft}(u, \varsigma)$ *[45], [46]:*

$$o_t = [soft(u, \varsigma)]_t = \begin{cases} u_t - \varsigma, & if\ \ u_t > \varsigma, \\ 0, & Otherwise, \end{cases} \tag{15}$$

where the threshold value ς can be calculated as in Step 4 of Algorithm 3.

Algorithm 3. Moreau Projection $S_\tau(G)$

```
Given input $G = [G^1, G^2, \ldots, G^\kappa]$ and $s = \frac{1}{\tau}$.
1: Calculate $\hat{u}_t = \|G^t\|_F$ for all $t = 1, \ldots, \kappa$.
2: Sort $\hat{u}$ to obtain $u$ such that $u_{(1)} \geq \cdots \geq u_{(\kappa)}$.
3: Find $\rho = \max\{t | u_t - \frac{s}{1+ts}\sum_{i=1}^{t} u_i > 0, t = 1, \ldots, \kappa\}$.
4: Calculate the threshold value $\varsigma = \frac{s}{1+\rho s}\sum_{i=1}^{\rho} u_i$.
5: Compute $o = soft(\hat{u}, \varsigma)$.
6: Compute and output $S_\tau(G)$.
```

The overall APG algorithm for solving problem (11) is summarized in Algorithm 4, where $F(\Omega) = \frac{\gamma}{2}(\sum_{t=1}^{C}\|\Omega^t\|_F)^2 + \frac{1}{2}\|\Xi\|_F^2$. Interested readers can find more details and the convergence derivation of Algorithm 1 and Algorithm 4 in [44].

3.3 Handling High Dimensional Sparse Problems

Given an ultrahigh dimensional sparse data matrix, removing the data mean (zero-centering) could make the matrix very dense. The data matrix $\binom{X}{\mathbf{1}_{1\times n}}$ can be used instead for regression to remove the data offset. As for the proposed framework, zero-centering can be performed in each subproblem. Zero-centering could also affect the computation of some regression responses T, such as PCA as in Table I, which assumes zero mean. In this case, $XX^\top$ can be first computed and centering can then be applied to both rows and columns as follows:

$$X\left(I - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top\right), \tag{16}$$

$$S = \left(I - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top\right)XX^\top\left(I - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top\right). \tag{17}$$

[1] In practice, the optimization is conducted on those selected features only.

TABLE 2
Computational Complexity of the Proposed Framework

Modules	Cholesky & QR decompositions	SVD	Finding p^t	Subproblem
Details	$S = HH^\top$, $H = QR$	$U_R \Sigma_R V_R^\top$	Compute XV, s in (10), and sort s	regression
Complexity	$O(nr^2)$	$O(r^3)$	$O(ndr + dr$ $+ m \log d)$	$O(m\kappa nr)$

Algorithm 4. Accelerated Proximal Gradient for Solving Problem (11)

```
Initialization: Initialize the Lipschitz constant $L_t = L_{t-1}$ and set $\Omega^{-1} = \Omega^0$ by warm start, $\tau_0 = L_t$, $\eta \in (0,1)$, parameter $\varrho^{-1} = \varrho^0 = 1$, and $k = 0$.
1: Set $V^k = \Omega^k + \frac{\varrho^{k-1}-1}{\varrho^k}(\Omega^k - \Omega^{k-1})$.
2: Set $\tau = \eta\tau_k$.
   Repeat
     Set $G = V^k - \frac{1}{\tau}\nabla f(V^k)$, compute $S_\tau(G)$.
     if $F(S_\tau(G)) \leq Q(S_\tau(G), V^k)$,
        set $\tau_k = \tau$, stop, break;
     else
        $\tau = \min\{\eta^{-1}\tau, L_t\}$.
     End
   Until convergence $F(S_\tau(G)) \leq Q(S_\tau(G), V^k)$
3: Set $\Omega^{k+1} = S_{\tau_k}(G)$.
4: Let $\varrho^{k+1} = \frac{1+\sqrt{(1+4(\varrho^k)^2)}}{2}$. Let $k = k+1$.
5: Quit if the stopping condition is achieved. Otherwise, go to step 1.
6: Let $L_t = \eta^2\tau_k$ and return.
```

3.4 Computational Complexity

There are a few components in the proposed framework. Given $S \in \mathbb{R}^{n\times n}$ with r significant singular values, the proposed framework requires a Cholesky decomposition $S = HH^\top$ and a QR-decomposition of $HP = QR$, SVD of R, finding p^t, and solving the subproblem. The decomposed matrices are compact, i.e., $Q \in \mathbb{R}^{n\times r}$, $R \in \mathbb{R}^{r\times r}$. Table 2 shows the computational complexity of different components. For $r \leq n$ and the desired feature cardinality $m\kappa \ll d$, the overall computational complexity is $O(ndr)$. This is much more efficient than FPS [10], whose complexity is $O(d^3 + nd^2)$ especially for high dimensional data applications where $d \gg m\kappa$, $d \gg n$, and $d \gg r$.

4 Experiments

In the experiments, we evaluated the proposed framework for unsupervised, supervised, and semi-supervised graph embedding, including PCA, LPP, LDA, and semi-supervised discriminant analysis (SDA). We termed them as SparGE-PCA, SparGE-LPP, SparGE-LDA, and SparGE-SDA, respectively, where SparGE stands for the proposed sparse graph embedding. We compared them with some current state-of-the-art algorithms surveyed in Section 1. A series of experiments on a wide range of datasets were conducted to compare the proposed methods with the current state-of-the-art algorithms.

TABLE 3
Datasets Used to Compare the SPCA, FPS, and SparGE-PCA Methods

S/N	Data	# dimensions	# instances
1	Toy data	550	5,000
2	Ramaswamy data	16,063	144

4.1 Experiments on Unsupervised Sparse Embedding

The proposed SparGE-PCA method chooses feature subsets that maximize the variance in the data in an unsupervised manner. It optimizes over all principal components simultaneously. On the other hand, most of the sparse PCA methods [41], [47], [48], [49] select features sequentially over the principal components (PC). It is assumed that the feature subset derived from the first PC should be more important, but this may not be true. A simple counterexample is to find one feature explaining the most variance of the covariance matrix $[3.2\ 0\ 0; 0\ 3\ 3; 0\ 3\ 3]$. In this case, SPCA [41] will select either the second or the third feature based on the first PC, but the correct selection should actually be the first feature. Only the recent concurrent work on Fantope Projection and Selection (FPS) by Vu et al. [10] shares a similar optimization objective as our proposed method.

To compare the proposed SparGE-PCA method with FPS and SPCA, we conducted experiments on both simulated data and a real gene data dataset shown in Table 3. The percentage of explained variance r_Σ was used to measure the quality of the selected feature set as follows:

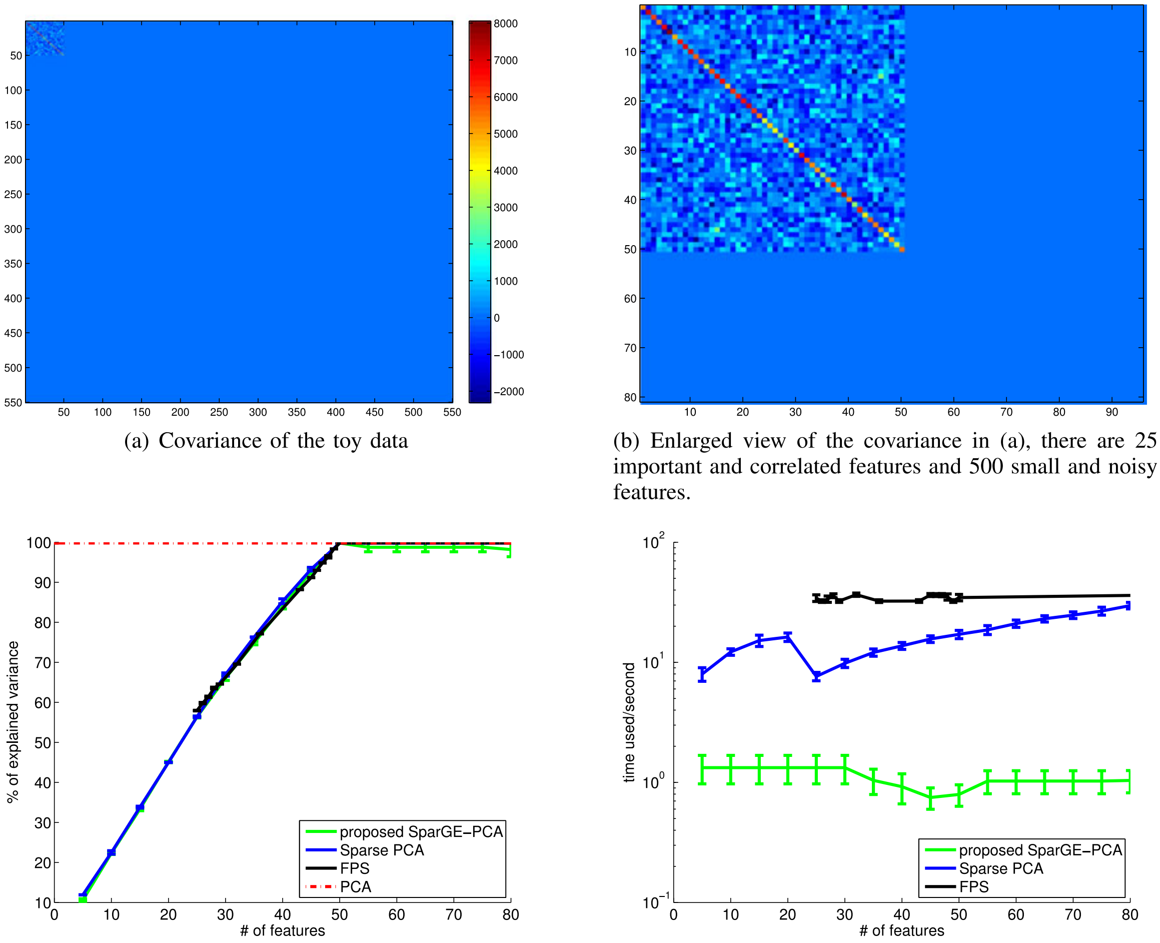
$$r_\Sigma = \frac{\text{trace}(\Sigma_{\text{fs}})}{\text{trace}(\Sigma)} \times 100\%, \tag{18}$$

where Σ_{fs} and Σ are the covariances of the selected features and of all features, respectively. A larger r_Σ indicates a better feature subset. The results of 50 independent runs are reported. For SPCA, we chose the features with the highest absolute magnitudes of the weight matrix, similar to [32].

4.1.1 The Results on Simulated Data

Optimizing for global variance, a good sparse embedding method should be able to identify a feature subset explaining the most variance and also removing noisy features. To test the optimality of the selected features of the proposed SparGE-PCA method, a simulation experiment is carried out. In this experiment, a toy data is generated with 50 significant features and 500 noisy features. Its covariance is shown in Fig. 2a. The explained variance should converge at 50 features. Non-zero values in the off-diagonal entries indicate a correlation among the first 50 significant features.

As shown in Fig. 2c, FPS, SPCA, and the proposed SparGE-PCA method all converged to the optimal variance at 50 features in accordance to the groundtruth. Computationally, SPCA was an order of magnitude slower than the proposed method, as shown in Fig. 2d. FPS shared a similar computational time with SPCA.



(a) Covariance of the toy data

(b) Enlarged view of the covariance in (a), there are 25 important and correlated features and 500 small and noisy features.

(c) Variance explained

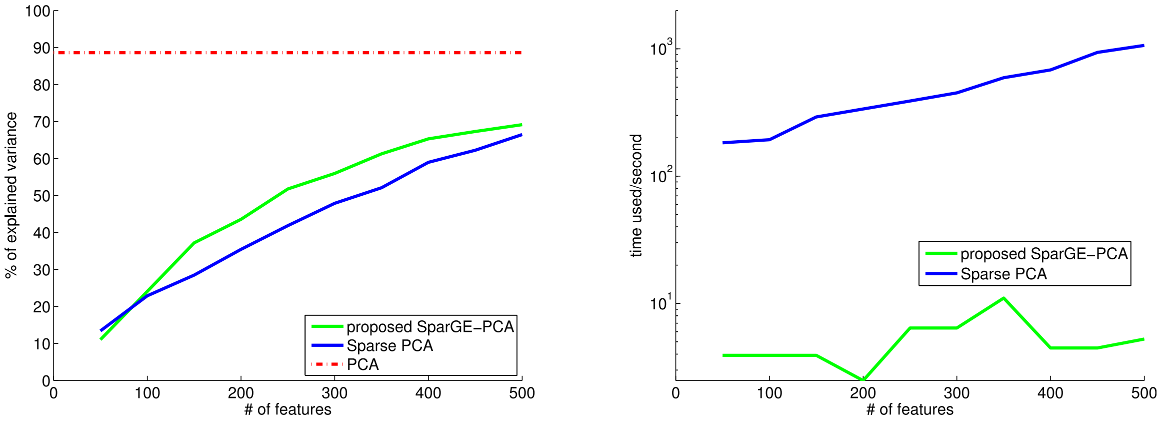
(d) Computational time

Fig. 2. Toy experiments to show the explained variance and run time versus features cardinality. The proposed SparGE-PCA method performed similarly to SPCA and FPS in the explained variance on a toy dataset with the first 80 important features, but it was at least an order of magnitude faster than SPCA and FPS. PCA with 25 subspaces using all of the dimensions explains about 99.8 percent of the total variance.

4.1.2 The Results on Real Data Sets

PCA is often used for analyzing high dimensional data with small samples, especially biological data. In this experiment, the microarray data, Ramaswamy data [41], is used to find the meaningful genes from very high dimensional data. The data has a very high dimensionality of 16,063 (genes) and 144 samples only. Only the first 25 principal components were used to select features.

In this experiment, the FPS method was unable to handle such a high dimensionality since the its computational



(a) Variance explained

(b) Computational time

Fig. 3. Comparing to the SPCA method on the Ramaswamy data for variance and computational time, the proposed SparGE-PCA method outperformed the SPCA method in the explained variance and was much more efficient. PCA with 25 significant subspaces explains about 89 percent of the total variance.

complexity is $O(d^3 + nd^2)$ when performing a Fantope Projection. Therefore, only the SparGE-PCA and SPCA methods were chosen for comparison.

The proposed SparGE-PCA method outperformed SPCA significantly in the explained variance, by about 10 percent between 200 and 350 features as shown in Fig. 3a. Both SPCA and the proposed SparGE-PCA method converged to 70 percent in the explained variance with 500 features (only about 3.1 percent of the total features). Computationally, SPCA was at least two orders of magnitude slower than the proposed method as shown in Fig. 3b.

4.2 Experiments on Feature Selection for Clustering

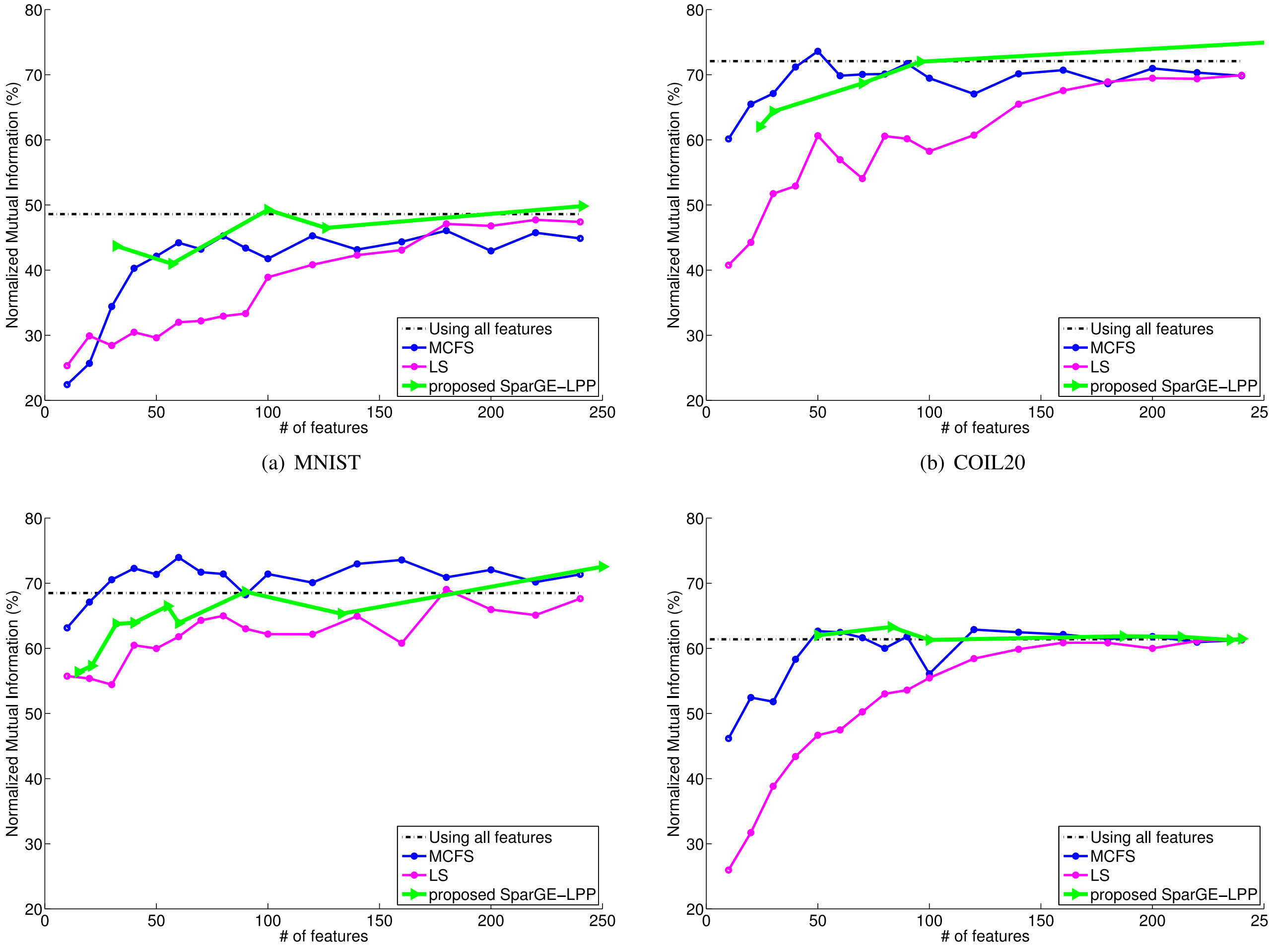
Besides sparse graph embedding for PCA, our proposed framework can be used to identify important features for clustering tasks. As discussed in Section 1, unsupervised LPP can model the local data structure well, and thus its embedding could improve clustering performance. Since image datasets, such as digits and faces images, usually lie on a low dimensional manifold, four popular image datasets, namely MNIST, COIL20, ORL, and USPS as shown in Table 4 were chosen.

To evaluate the quality of selected features, we apply k-means clustering on the data with the chosen features, where k is set to the number of classes. The normalized mutual information defined in [21] is used as the performance measure. The clustering baseline employed all of the features. Besides the baseline, the feature ranking methods such as Laplacian Scores (LS) and MultiClusters Feature Selection (MCFS) were also chosen for a comparative evaluation.

TABLE 4
Image Datasets used for Clustering

S/N	Data	# dimensions	# instances	# classes
1	MNIST	784	4,000	10
2	COIL20	1,024	1,440	20
3	ORL	1,024	400	40
4	USPS	256	9,298	10

The result was shown in Fig. 4 with up to 250 features only. Interestingly, the baseline results could be achieved with as few as 50 to 70 features. Both MCFS and the proposed SparGE-LPP method performed much better than the LS feature selection method. The proposed SparGE-LPP method also outperformed MCFS on MNIST, COIL20, and USPS. Note that MCFS employed a simple ranking method to choose features from a sequential learning framework. Theoretically, it is unclear whether this approach could converge to optimality. Furthermore,



(a) MNIST

(b) COIL20

(c) ORL 32x32

(d) USPS

Fig. 4. Comparison of different feature selection methods on clustering tasks.

it does not learn the collective weights based on the common feature subsets. In contrast, the proposed framework could achieve both weight learning and feature subset selection simultaneously.

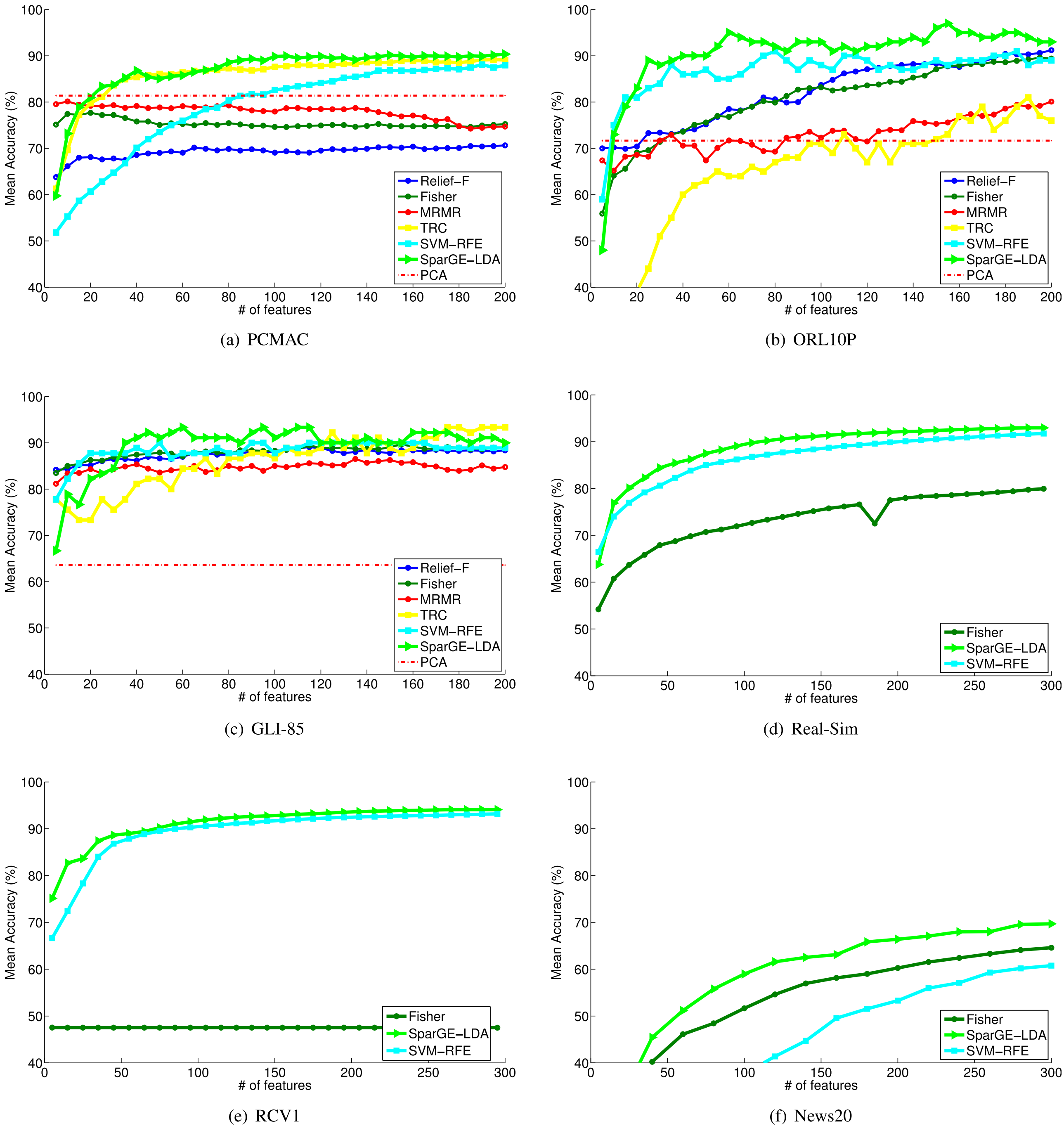
4.3 Experiments on Supervised Feature Selection

In this section, we compare the proposed SparGE-LDA method with some chosen feature selection methods on classification tasks. Six datasets as shown in Table 5 are used for comparison. The first three datasets ranging from text, images, and microarray data, have medium dimension and small sample sizes, and are collected from the Arizona State University (ASU) feature selection repository [50]. The

TABLE 5
A Set of Data used for Classification Experiments

S/N	Data	# dimensions	# instances	# classes	Type
1	PCMAC	3,289	1,943	2	Text
2	ORL10P	10,304	100	10	Face image
3	GLI-85	22,283	85	2	Microarray data
4	Real-Sim	20,958	72,309	2	Text
5	RCV1	47,236	training:20,242, testing: 67,739	2	Text
6	News20	62,060	training:15,935, testing: 3,993	20	Text

Unless indicated otherwise, datasets were split for 10-fold cross validation.



(a) PCMAC

(b) ORL10P

(c) GLI-85

(d) Real-Sim

(e) RCV1

(f) News20

Fig. 5. Comparison of different feature selection methods on classification tasks. Classification results of PCA as pre-processing is also included as a benchmark for the first three datasets.

TABLE 6
Computational Time in Seconds of SparGE-LDA, the Fisher Score Method, and SVM-RFE on Ultrahigh Dimensional Data

Methods	Real_Sim	RCV1	News20
SparGE-LDA	228	302	460
Fisher	53	91	329
SVM-RFE	155	344	24,160

other three higher dimensional and large-scale text datasets are from the LIBSVM [51] repository.

Several popular feature selection methods, such as Relief-F [52], max-dependency Max-Relevance and Min-Redundancy (MRMR) [53], Fisher score, and SVM Recursive Feature Elimination (SVM-RFE) are chosen for comparison. Beside these, we also compared the most recent method by Nie et al. [23], the Trace Ratio Criterion (TRC) method. TRC was shown to outperform many methods in the literature. PCA is also included as a benchmark for classification using SVM. However, PCA cannot handle datasets such as Real-Sim, RCV1, and News20, which have both ultrahigh dimensions and large data sample sizes. Therefore, no comparison with PCA for these three datasets is included. For RCV1 and News20 datasets, the accuracies on the test sets are reported. For the rest, the mean accuracies of 10-fold cross-validation are reported.

The results are shown in Fig. 5. We do not report the results of Relief-F, TRC, and MRMR on the three high dimensional and large-scale datasets because these methods are computationally inefficient on these datasets. Generally, both TRC and the SparGE-LDA method can handle global feature subset directly, they outperformed the feature-level selection processes such as MRMR, Relief-F, and Fisher score methods. Compared with TRC, our SparGE-LDA outperformed for the ORL10P and GLI-85 significantly, was marginally better than TRC for the PCMAC dataset. In the TRC method, the optimization enforces only one feature in each column of the weight matrix; it is thus more constrained and may thus have a poorer performance.

On the other hand, the SparGE-LDA method outperformed SVM-RFE for the ORL10P, PCMAC, Real-Sim, RCV1, and News20 datasets. On the first three small datasets, SVM-RFE performed well too as it directly optimized for the classification methods. However, SVM-RFE is a greedy method, and its performance dropped significantly in the higher dimensional and large-scale data shown in Fig. 5d, 5e, and 5f. Compared to SVM-RFE, SparGE-LDA achieved 20 percent better in performance on the News20 dataset, 5 percent on the Real-Sim dataset, and 10 percent on the RCV1 dataset. On the other hand, the Fisher score method, the counterpart of SparGE-LDA, significantly underperformed on the last three high dimensional datasets.

Computationally, the Fisher score method was the fastest, but SparGE-LDA was also efficient and completed the tasks within minutes as shown in Table VI. SVM-RFE was generally fast, but it was very slow on the News20 dataset, which had the highest dimension and number of classes among the three ultrahigh dimensional datasets.

4.4 Experiments on Semi-Supervised Feature Selection

So far, the experiments demonstrated the effectiveness of our proposed framework for both unsupervised and supervised learning settings. By incorporating small labeled samples, the semi-supervised discriminant analysis based on the proposed framework (SparGE-SDA) can achieve a good classification rate.

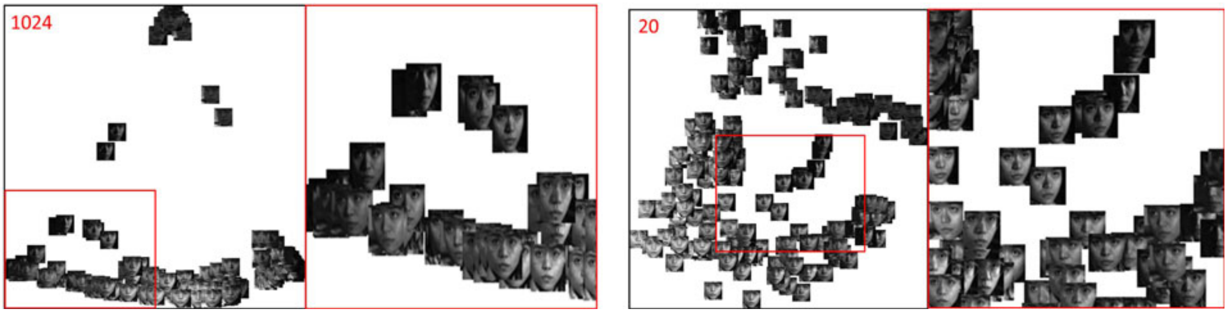
Two datasets (COIL and ORL) from Table 4 are used for evaluation. The proposed SparGE-SDA method was compared with the SVM-RFE and Fisher score feature selection methods. The classification accuracies are tabulated in Table 7. The proposed SparGE-SDA method was much better than the Fisher score method and consistently better than SVM-RFE.

4.5 Experiments on Data Visualization

In this section, we intend to identify the important face features that can preserve the data locality, i.e., the neighborhood relationships of these image samples. The data locality is visualized using two dimensional linear embedding of face images by LPP. From the CMU-PIE database [54], 170 face images of one person shown in Fig. 1a were chosen for this experiment. The linear embedding of these images is shown in Fig. 6. In Fig. 6a, it can be observed that the illumination increases from left to right, and her face turns from left to right. The faces with expressions are far apart from the rest showing on the top, indicating a large difference. A similar observation can be drawn using the proposed SparGE-LPP with only 20 features. Variations in illumination, poses, and expressions are much more gradual on the

TABLE 7
Feature Selection Results for the SparGE-SDA, SVM-RFE and Fisher Score Methods

		1 labeled samples			2 labeled samples			3 labeled samples		
Datasets	**# feats**	SparGE-SDA	SVM-RFE	Fisher	SparGE-SDA	SVM-RFE	Fisher	SparGE-SDA	SVM-RFE	Fisher
COIL	100	**59.7**	56.8	31.9	**64.3**	61.8	39.7	**71.3**	70.3	50.4
	200	**61.2**	57.6	37.0	**67.1**	62.6	54.9	**74.5**	70.8	61.1
	300	**64.2**	57.9	50.0	**69.1**	63.4	58.5	**76.7**	71.1	65.5
ORL	100	**59.0**	48.5	25.3	**65.3**	63.3	56.8	**70.1**	73.9	34.8
	200	**64.7**	52.2	40.0	**71.5**	66.5	64.9	**77.5**	75.9	68.9
	300	**65.0**	53.9	48.5	**71.9**	66.6	69.2	76.8	**76.9**	74.2



(a) Linear embedding of face images using LPP on all of the pixels

(b) Linear embedding of face images using the proposed SparGE-LPP method on 20 pixels

Fig. 6. Linear embedding of face images using all of the pixels and the 20 selected pixels by the proposed SparGE-LPP method. The proposed SparGE-LPP method has a better embedding as indicated by a gradual change in different poses, illumination, and expressions. The respective enlarged portions are shown in the red boxes.

embedding shown in Fig. 6b, indicating a better fit of the underlying manifold.

5 Conclusion

This paper proposes a novel unified framework to select features for generalized graph embedding. It utilizes a feature selector to directly optimize feature subsets for graph embedding in modeling the intrinsic data structures, enabling a more robust embedding, especially for high dimensional data with a small sample size. Its efficiency and effectiveness have been demonstrated with a series of experiments for clustering, classification, and visualization. In the experiments, the proposed methods outperformed the current state-of-the-art algorithms for unsupervised, supervised, and semi-supervised learning tasks. The proposed framework demonstrated its computational and memory efficiency in handling ultrahigh dimensional data for classification.

Appendix A: Proof of Proposition 1

In Proposition 1, the dual form of problem (11) has the same form as in problem (9).

Let $\Omega = [\Omega^1, \Omega^2, \ldots, \Omega^\kappa]$ be the stack of Ω^t. Define the cone $\mathcal{Q} = \{(\Omega, v)\} \in (\mathbb{R}^{d\times r}, \mathbb{R})|\|\Omega\|_F \le v\}$. Let $z_t = \|\Omega^t\|_F$, $z = \sum_{t=1}^{\kappa} z_t$. The optimization problem in (11) is equivalent to the following problem:

$$\begin{aligned} \min_{z,\Omega}\ & \frac{\lambda}{2} z^2 + \frac{1}{2}||\Xi||_F^2, \\ \text{s.t.}\ & \Xi = T - \sum_{t=1}^{\kappa} X^\top \mathrm{diag}(p^t)\Omega^t, \\ & \sum_{t=1}^{\kappa} z_t \le z,\quad (\Omega^t, z_t) \in \mathcal{Q}_r. \end{aligned} \tag{19}$$

The Lagrangian function of (19) can be written as:

$$\mathcal{L} = \frac{\gamma}{2} z^2 + \frac{1}{2}\|\Xi\|_F^2 - tr\left(V^\top\left(\sum_{t=1}^{\kappa} X^\top \mathrm{diag}(p^t)\Omega^t - T + \Xi\right)\right) + \eta\left(\sum_{t=1}^{\kappa} z_t - z\right) - \sum_{t=1}^{\kappa}\left(tr((\xi^t)^\top\Omega^t) + \mu_t z_t\right),$$

where $V \in \mathbb{R}^{n\times r}$, $\eta \in \mathbb{R}$, $\xi^t \in \mathbb{R}^{n\times r}$, and $\mu_t \in \mathbb{R}$ are the Lagrangian dual variables for the corresponding constraints. By setting the derivatives of $\mathcal{L}$ w.r.t. z, z_t, Ω^t, and Ξ to zero, we obtain the KKT conditions as follows:

$$\gamma z = \eta = \mu_t, \xi^t = -X\mathrm{diag}(p^t)V, \Xi = -V, \|\xi^t\|_F \le \eta.$$

Substitute these results into the Lagrangian function, and we obtain the dual problem as follows:

$$\max_{V,\eta}\ tr(V^\top T) - \frac{1}{2}tr(V^\top V) - \frac{1}{2\gamma}\eta^2 \tag{20}$$

$$\text{s.t.}\quad \|X\,\mathrm{diag}(p^t)V\|_F \le \eta, t = 1, \ldots, k. \tag{21}$$

Setting $\theta = tr(V^\top T) - \frac{1}{2}tr(V^\top V) - \frac{1}{2\gamma}\eta^2$ and $f(V, p^t) = tr(V^\top T) - \frac{1}{2\gamma}\|X\mathrm{diag}(p^t)V\|_F^2 - \frac{1}{2}tr(V^\top V)$, then the problem becomes as follows:

$$\begin{aligned} \max_{V\in\mathbb{R}^{n\times k}, \eta\in\mathbb{R}}\ & \theta \\ \text{s.t.}\ & \theta \le f(V, p^t), \quad \forall p^t \in \Pi. \end{aligned} \tag{22}$$

Acknowledgments

The authors would like to thank Arizona State University for sharing the feature selection datasets and algorithms in its repositories and the anonymous reviewers for their valuable and detailed suggestions to improve the quality of the paper. This work was supported by the Australian Research Council Future Fellowship FT130100746. Marcus Chen is a corresponding author of the article.

References

[1] D. L. Donoho, “High-dimensional data analysis: The curses and blessings of dimensionality,” *AMS Math Challenges Lecture*, pp. 1–32, 2000.

[2] Y. Zhai, Y. Ong, and I. Tsang, “The emerging “big dimensionality”, ”*IEEE Comput. Intell. Mag.*, vol. 9, no. 3, pp. 14–26, Jul. 2014.

[3] R. Bellman and R. Bellman. (1957). *Dynamic Programming*. ser. P (Rand Corporation). Princeton, NJ, USA: Princeton University Press. [Online]. Available: http://books.google.com.sg/books?id=rZW4ugAACAAJ

[4] X. He and P. Niyogi, “Locality preserving projections,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 16, 2004, p. 153.

[5] S. Yan, D. Xu, B. Zhang, H. Zhang, Q. Yang, and S. Lin. (2007, Jan.). Graph embedding and extensions: A general framework for dimensionality reduction. *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 1, pp. 40–51. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4016549

[6] G. McLachlan, *Discriminant Analysis and Statistical Pattern Recognition*. New York, NY, USA: Wiley, vol. 544, 2004.

[7] J. B. Tenenbaum, V. De Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

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

[9] I. M. Johnstone and A. Y. Lu, "On consistency and sparsity for principal components analysis in high dimensions," *J. Am. Statist. Assoc.*, vol. 104, no. 486, pp. 682–693, 2009.

[10] V. Q. Vu, J. Cho, J. Lei, and K. Rohe, "Fantope projection and selection: A near-optimal convex relaxation of sparse pca," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 2670–2678.

[11] Y. Zhai, M. Tan, I. W. Tsang, and Y.-S. Ong, "Discovering support and affiliated features from very high dimensions," in *Proc. Int. Conf. Mach. Learn.*, 2012, pp. 1455–1462.

[12] Q. Mao and I.-H. Tsang, "A feature selection method for multivariate performance measures," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 9, pp. 2051–2063, Sep. 2013.

[13] M. Dash and H. Liu, "Feature selection for classification," *Intell. Data Anal.*, vol. 1, no. 1–4, pp. 131–156, 1997.

[14] M. J. Martin-Bautista and M.-A. Vila, "A survey of genetic feature selection in mining issues," in *Proc. IEEE Evolutionary Comput., Congr.*, vol. 2, 1999, pp. 13–21.

[15] C.-F. Tsai. (2009, Mar.). Feature selection in bankruptcy prediction. *Knowl.-Based Syst.*, vol. 22, no. 2, pp. 120–127. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0950705108001536

[16] Y. Saeys, I. Inza, and P. Larranaga. (2007, Oct.). A review of feature selection techniques in bioinformatics. *Bioinformatics*, vol. 23, no. 19, pp. 2507–2517. [Online]. Available: http://bioinformatics.oxfordjournals.org/cgi/doi/10.1093/bioinforma tics /btm344

[17] H. Liu and L. Yu. (2005, Apr.). Toward integrating feature selection algorithms for classification and clustering. *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 4, pp. 491–502. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1401889

[18] Q. Gu, Z. Li, and J. Han, "Generalized Fisher score for feature selection," in *Proc. 27th Conf. Uncertainty Artif. Intell.*, 2011, pp. 266–273.

[19] M. A. Hall, "Correlation-based feature selection for machine learning," Ph.D. dissertation, Univ. Waikato, Hamilton, New Zealand, 1999.

[20] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, 2003.

[21] X. He, D. Cai, and P. Niyogi, "Laplacian score for feature selection," *Adv. Neural Inf. Process. Syst.*, vol. 18, p. 507, 2006.

[22] Z. Zhao and H. Liu, "Spectral feature selection for supervised and unsupervised learning," in *Proc. 24th Int. Conf. Mach. Learn.*, 2007, pp. 1151–1157.

[23] F. Nie, S. Xiang, Y. Jia, C. Zhang, and S. Yan, "Trace ratio criterion for feature selection," in *Proc. 23rd Natl. Conf. Artif. Intell.*, vol. 2, 2008, pp. 671–676.

[24] H. Liu, H. Motoda, R. Setiono, and Z. Zhao, "Feature selection: An ever evolving frontier in data mining." *J. Mach. Learn. Res.-Proc. Track*, vol. 10, pp. 4–13, 2010.

[25] C. Hou, F. Nie, X. Li, D. Yi, and Y. Wu, "Joint embedding learning and sparse regression: A framework for unsupervised feature selection," *IEEE Trans. Cybern.*, vol. 44, no. 6, pp. 793–804, Jun. 2013.

[26] X. Cai, F. Nie, and H. Huang, "Exact top-k feature selection via l2,0-norm constraint," in *Proc. Twenty-Third Int. Joint Conf. Artif. Intell.*, 2013, pp. 1240–1246.

[27] F. Nie, H. Huang, X. Cai, and C. Ding, "Efficient and robust feature selection via joint l2,1-norms minimization," *Adv. Neural Inf. Process. Syst.*, vol. 23, pp. 1813–1821, 2010.

[28] S. Xiang, F. Nie, G. Meng, C. Pan, and C. Zhang, "Discriminative least squares regression for multiclass classification and feature selection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 11, pp. 1738–1754, Oct. 2012.

[29] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Royal Statist. Soc. Series B (Methodological)*, vol. 58, pp. 267–288, 1996.

[30] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *J. Royal Statist. Soc.: Series B (Statistical Methodology)*, vol. 67, no. 2, pp. 301–320, 2005.

[31] H. Zou, "The adaptive lasso and its oracle properties," *J. Am. Statist. Associat.*, vol. 101, no. 476, pp. 1418–1429, 2006.

[32] D. Cai, C. Zhang, and X. He, "Unsupervised feature selection for multi-cluster data," in *Proc. 16th ACM SIGKDD Int. Conf. Knowledge Discovery Data Min.*, 2010, pp. 333–342.

[33] F. Bach, S. D. Ahipasaoglu, and A. d'Aspremont, "Convex relaxations for subset selection," *arXiv preprint arXiv:1006.3601*, 2010.

[34] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, "Least angle regression," *Ann. statist.*, vol. 32, no. 2, pp. 407–499, 2004.

[35] L. Sun, S. Ji, and J. Ye, "A least squares formulation for a class of generalized eigenvalue problems in machine learning," in *Proc. 26th Annual Int. Conf. Mach. Learn.*, 2009, pp. 977–984.

[36] D. Cai, X. He, and J. Han, "Semi-supervised regression using spectral techniques," *Dept. Comput. Sci., Univ. Illinois Urbana-Champaign, Urbana, IL, USA, Tech. Rep. UIUCDCS*, 2006.

[37] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *J. Mach. Learn. Res.*, vol. 7, pp. 2399–2434, 2006.

[38] X. Zhu, "Semi-supervised learning literature survey," *Compute. Sci., Univ. Wisconsin-Madison, Madison, WI, USA*, vol. 2, p. 3, 2006.

[39] Y. Liu, F. Nie, J. Wu, and L. Chen, "Efficient semi-supervised feature selection with noise insensitive trace ratio criterion," *Neurocomputing*, vol. 105, pp. 12–18, 2013.

[40] D. Cai, X. He, and J. Han, "Semi-supervised discriminant analysis," in *Proc. Int. Conf. Comput. Vis.*, 2007, pp. 1–7.

[41] H. Zou, T. Hastie, and R. Tibshirani, "Sparse principal component analysis," *J. Comput. Graph. Statist.*, vol. 15, no. 2, pp. 265–286, 2006.

[42] M. Tan, L. Wang, and I. W. Tsang, "Learning sparse svm for feature selection on very high dimensional datasets," in *Proc. 27th Int. Conf. Mach. Learn.*, 2010, pp. 1047–1054.

[43] F. R. Bach, G. R. Lanckriet, and M. I. Jordan. (2004). Multiple kernel learning, conic duality, and the smo algorithm. in *Proc. Twenty-first Int. Conf. Mach. Learn.*, New York, NY, USA: ACM Press, p. 6. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1015330.1015424

[44] M. Tan, I. W. Tsang, and L. Wang, "Towards ultrahigh dimensional feature selection for big data," *J. Mach. Learning Research*, vol. 15, pp. 1371–1429, 2014.

[45] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations Trends Optim.*, vol. 1, no. 3, pp. 123–231, 2013.

[46] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Commun. Pure Appl. Math.*, vol. 57, no. 11, pp. 1413–1457, 2004.

[47] M. Hein and T. Bühler, "An inverse power method for nonlinear eigenproblems with applications in 1-spectral clustering and sparse pca," *arXiv preprint arXiv:1012.0774*, 2010.

[48] M. Journée, Y. Nesterov, P. Richtárik, and R. Sepulchre, "Generalized power method for sparse principal component analysis," *J. Mach. Learn. Res.*, vol. 11, pp. 517–553, 2010.

[49] X.-T. Yuan and T. Zhang, "Truncated power method for sparse eigenvalue problems," *arXiv preprint arXiv:1112.2679*, 2011.

[50] Z. Zhao, F. Morstatter, S. Sharma, S. Alelyani, A. Anand, and H. Liu, "Advancing feature selection research," *ASU Feature Selection Repository*, 2010.

[51] C.-C. Chang and C.-J. Lin, "Libsvm: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, p. 27, 2011.

[52] H. Liu, and H. Motoda, *Computational Methods of Feature Selection*. Boca Raton, FL, USA: Chapman and Hall/CRC, 2007.

[53] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1226–1238, Jun. 2005.

[54] T. Sim, S. Baker, and M. Bsat, "The CMU pose, illumination, and expression (pie) database," in *Proc. Fifth IEEE Int. Conf. Autom. Face Gesture Recognit.*, 2002, pp. 46–51.

Marcus Chen received the Bachelor of Science degree in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, in 2007, and the Master of Science degree in electrical engineering from Stanford University, Stanford, CA, in 2008. He is currently working toward the PhD degree at Nanyang Technological University, Singapore. His research interests include pattern recognition, optimization, large scale image search, and remote sensing.

Ivor W. Tsang received the PhD degree in computer science from the Hong Kong University of Science and Technology, Hong Kong, in 2007. He is an Australian Future Fellow and Associate Professor with the Centre for Quantum Computation & Intelligent Systems (QCIS), at the University of Technology, Sydney (UTS), Sydney, Australia. Before joining UTS, he was the Deputy Director of the Centre for Computational Intelligence, Nanyang Technological University, Singapore. He has published more than 100 research papers in referred international journals and conference proceedings, including *JMLR, TPAMI, TNN/TNNLS,* NIPS, ICML, UAI, SIGKDD, IJCAI, AAAI, ACL, ICCV, and CVPR. In 2009, he was conferred the 2008 Natural Science Award (Class II) by the Ministry of Education, China, which recognized his contributions to kernel methods. In 2013, he received the prestigious Australian Research Council Future Fellowship for his research regarding Machine Learning on Big Data. In addition, he received the prestigious IEEE Transactions on Neural Networks Outstanding 2004 Paper Award in 2006, the 2014 IEEE Transactions on Multimedia Prized Paper Award, and a number of best paper awards and honors from reputable international conferences, including the Best Student Paper Award at CVPR 2010, the Best Paper Award at ICTAI 2011 and the Best Poster Award Honorable Mention at ACML 2012. He was also awarded the Microsoft Fellowship 2005 and the ECCV 2012 Outstanding Reviewer Award.

Mingkui Tan received the Bachelor's degree in environmental science and engineering and the Master's degree in control science and engineering, in 2006 and 2009, respectively, both from Hunan University in Changsha, China. He received the PhD degree in computer science from Nanyang Technological University, Singapore, in 2014. He is currently a Senior Research Associate with the School of Computer Science at The University of Adelaide, Adelaide, Australia. His research interests include compressive sensing, big data learning, and large-scale optimization.

Tat Jen Cham received the Bachelor's degree in engineering and the PhD degree, in 1993 and 1996, respectively, both from the University of Cambridge, Cambridge, United Kingdom. He is currently an Associate Professor in the School of Computer Engineering at Nanyang Technological University, Singapore, the Director of the Centre for Multimedia & Network Technology (CeMNet), and a key researcher in the BeingThere Centre for 3D Telepresence. His research interests are broadly in computer vision and machine learning. He was also a research scientist in the DEC / Compaq Cambridge Research Lab in Boston, MA, from 1998 to 2001.