Sparse Projections over Graph

Deng Cai
Computer Science Department
University of Illinois at Urbana-Champaign
dengcai2@cs.uiuc.edu

Xiaofei He
College of Computer Science
Zhejiang University, China
xiaofeihe@cad.zju.edu.cn

Jiawei Han
Computer Science Department
University of Illinois at Urbana-Champaign
hanj@cs.uiuc.edu

Abstract

Recent study has shown that canonical algorithms such as Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) can be obtained from graph based dimensionality reduction framework. However, these algorithms yield projective maps which are linear combination of *all* the original features. The results are difficult to be interpreted psychologically and physiologically. This paper presents a novel technique for learning a sparse projection over graphs. The data in the reduced subspace is represented as a linear combination of a subset of the most relevant features. Comparing to PCA and LDA, the results obtained by sparse projection are often easier to be interpreted. Our algorithm is based on a graph embedding model, which encodes the discriminating and geometrical structure in terms of the data affinity. Once the embedding results are obtained, we then apply regularized regression for learning a set of sparse basis functions. Specifically, by using a L_1-norm regularizer (e.g. *lasso*), the sparse projections can be efficiently computed. Experimental results on two document databases demonstrate the effectiveness of our method.

Introduction

Dimensionality reduction has been a key problem in many fields of information processing, such as data mining, information retrieval, and pattern recognition. The most popular linear methods include Principal Component Analysis (PCA) (Duda, Hart, & Stork 2000) and Linear Discriminant Analysis (LDA) (Fukunaga 1990).

One of the major disadvantages of these two algorithms is that the learned projective maps are linear combinations of *all* the original features, thus it is often difficult to interpret the results. Recent psychological and physiological evidence have shown that the representation of objects in human brain may be component-based (Lee & Seung 1999). This motivates us to develop sparse subspace learning algorithms. In other words, the data in the reduced subspace is represented as a linear combination of a subset of the features which are the most informative. Zhou *et al*. (Zhou, Hastie, & Tibshirani 2004) proposed an elegant sparse PCA (SparsePCA) algorithm by using L_1-penalized regression on regular principle components, which can be solved very efficiently using *least angle regression* (LARS) (Efron *et al*. 2004). More recently, d'Aspremont *et al*. (d'Aspremont *et al*. 2004) relaxed the hard cardinality constraint and obtained a convex approximation by using semi-definite programming. In (Moghaddam, Weiss, & Avidan 2005; 2006), Moghaddam *et al*. proposed a spectral bounds framework for sparse subspace learning. Particularly, they proposed both exact and greedy algorithms for both sparse PCA and sparse LDA. It would be important to note that the sparse LDA algorithm (Moghaddam, Weiss, & Avidan 2006) can only be applied to two-class problems.

In this paper, we propose a novel algorithm for learning a Sparse Projection over Graphs (SPG). Recent study has shown that both PCA and LDA can be obtained from graph Laplacian based dimensionality reduction framework (He *et al*. 2005). Using techniques from spectral graph theory (Chung 1997), we construct an affinity graph to encode both discriminating and geometrical structure in the data. The affinity graph is usually sparse (e.g. nearest neighbor graph), so the embedding results can be very efficiently computed. Once we get the embedding results, *lasso* regression can be naturally applied to obtain sparse basis functions.

The new algorithm is interesting from a number of perspectives.

1. Comparing to canonical subspace learning algorithms such as PCA and LDA, our algorithm produces sparse basis functions which can be better interpreted psychologically and physiologically.
2. Comparing to previous sparse subspace learning algorithms such as sparse PCA (Zhou, Hastie, & Tibshirani 2004) and sparse LDA (Moghaddam, Weiss, & Avidan 2006), our algorithm is more flexible. To be specific, our algorithm can be performed in either supervised, unsupervised, or semi-supervised manner. When there is label information available, it can be easily incorporated into the graph structure.
3. Unlike sparse LDA (Moghaddam, Weiss, & Avidan 2006) which can only be applied to two-class problems, our algorithm can be applied to multiple-class problems.

Graph Laplacian based Linear Dimensionality Reduction

In this Section, we provide a brief review of graph Laplacian based dimensionality reduction.

Suppose we have m data samples $\{\mathbf{x}_i\}_{i=1}^m \subset \mathbb{R}^n$ and let $X = (\mathbf{x}_1, \cdots, \mathbf{x}_m)$. Consider a linear map $f(\mathbf{x}) = \mathbf{a}^T\mathbf{x}$. The optimal $\mathbf{a}$ can be obtained by minimizing the following loss function:

$$\sum_{i,j}(\mathbf{a}^T\mathbf{x}_i - \mathbf{a}^T\mathbf{x}_j)^2 W_{ij}$$

where W is the weight matrix of a given graph constructed over all the data points. Both discriminant and geometrical structure can be encoded into the graph and the resulting embedding results found by solving the above objection function respect the defined graph structure.

It would be interesting to note that recent study has shown that many popular linear dimensionality reduction algorithms can be derived from the graph Laplacian framework. Particularly, He *et al*. have shown that with specially designed graph structure, we can get both PCA and LDA (He *et al.* 2005):

Graph structure for PCA

$$W = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{pmatrix}$$

Graph structure for LDA

$$W = \begin{pmatrix} \frac{1}{m_1} & \cdots & \frac{1}{m_1} & 0 & \cdots & \cdots & 0 \\ \vdots & \ddots & \vdots & 0 & \cdots & \cdots & 0 \\ \frac{1}{m_1} & \cdots & \frac{1}{m_1} & 0 & \cdots & \cdots & 0 \\ 0 & \cdots & 0 & \ddots & 0 & \cdots & 0 \\ \vdots & \cdots & \cdots & 0 & \frac{1}{m_c} & \cdots & \frac{1}{m_c} \\ \vdots & \cdots & \cdots & 0 & \vdots & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 & \frac{1}{m_c} & \cdots & \frac{1}{m_c} \end{pmatrix}$$

where $m_i (i = 1, \cdots, c)$ is the number of data points in the i-th class. Clearly, the PCA graph describes the global geometrical structure, whereas the LDA graph describes the discriminant structure.

Sparse Projection over Graphs

In this section, we introduce our algorithm for learning a sparse projection over graphs. We begin with a formal description of the learning problem.

The Problem

The generic problem of linear sparse dimensionality reduction is the following. Given a set $\mathbf{x}_1, \cdots, \mathbf{x}_m$ in $\mathbb{R}^n$, find a transformation matrix $A = (\mathbf{a}_1, \cdots, \mathbf{a}_l)$ that maps these m points to a set of points $\mathbf{y}_1, \cdots, \mathbf{y}_m$ in $\mathbb{R}^l$ ($l \ll n$), such that $\mathbf{y}_i (= A^T\mathbf{x}_i)$ "represents" $\mathbf{x}_i$ and the cardinality of $\mathbf{a}_i$ ($i = 1, \cdots, l$) is less than k, where $k (< n)$ is a suitable integer. The cardinality of a vector is defined as the number of non-zero entries.

The Algorithm

We adopt graph Laplacian framework to develop an algorithm for learning a sparse projection. Given graph G with weight matrix W over the data points, we aim to minimize the following objective function:

$$\begin{array}{rl} \textbf{min} & \sum_{ij}\left(\mathbf{a}^T\mathbf{x}_i - \mathbf{a}^T\mathbf{x}_j\right)^2 W_{ij} \\ \text{subject to} & \mathbf{a}^T X D X^T \mathbf{a} = 1 \\ & \text{card}(\mathbf{a}) \leq k \end{array} \quad (1)$$

where D is a diagonal matrix whose entries are the row (or column, since W is symmetric) sums of W. That is, $D_{ii} = \sum_j W_{ij}$. Let $\mathbf{y}^i = (y_1^i, \cdots, y_m^i)$ be a projection on the eigenvector $\mathbf{a}_i$. For any $i \neq j$, it is easy to show that $\mathbf{y}^i D (\mathbf{y}^i)^T = \mathbf{y}^j D (\mathbf{y}^j)^T = 1$ and $\mathbf{y}^i D (\mathbf{y}^j)^T = 0$. This indicates that the projections in the reduced space are weighted uncorrelated.

The objective function (1) is NP-hard and therefore generally intractable. In the following we describe an efficient method for solving this optimization problem. By simple algebraic formulation, it is easy to verify:

$$\sum_{ij}\left(\mathbf{a}^T\mathbf{x}_i - \mathbf{a}^T\mathbf{x}_j\right)^2 W_{ij} = \mathbf{a}^T X L X^T \mathbf{a} \quad (2)$$

where the matrix $L = D - W$ is usually called graph Laplacian (Chung 1997). A natural relaxation to this problem is to first remove the cardinality constraint and solve the following eigenvector problem:

$$XLX^T\mathbf{a} = \lambda XDX^T\mathbf{a} \quad (3)$$

Once the embedding results $y_i = \mathbf{a}^T\mathbf{x}_i$ are obtained, we can apply lasso regression to get a sparse transformation vector. Specifically, let $\tilde{\mathbf{a}}$ be the sparse approximation of $\mathbf{a}$. Thus, $\tilde{\mathbf{a}}$ can be obtained by minimizing the sum of least squares with L_1-norm penalty:

$$\min_{\tilde{\mathbf{a}}}\left(\sum_{i=1}^m\left(\mathbf{a}^T\mathbf{x}_i - \tilde{\mathbf{a}}^T\mathbf{x}_i\right)^2 + \beta\sum_{j=1}^n|\tilde{a}_j|\right) \quad (4)$$

which is equivalent to

$$\min_{\tilde{\mathbf{a}}}\left(\sum_{i=1}^m\left(y_i - \tilde{\mathbf{a}}^T\mathbf{x}_i\right)^2 + \beta\sum_{j=1}^n|\tilde{a}_j|\right) \quad (5)$$

where $\tilde{a}_j$ is the j-th element in $\tilde{\mathbf{a}}$. Due to the nature of the L_1-norm penalty, some coefficients $\tilde{a}_j$'s will be shrunk to exact zero if β is sufficiently large. Specifically, for any given k, there exists β such that the solution of the optimization problem in Eqn. (5) satisfies $\text{card}(\tilde{\mathbf{a}}) \leq k$ (Hastie, Tibshirani, & Friedman 2001)(Efron *et al.* 2004). The *Least Angel Regression* (LARS) algorithm (Efron *et al.* 2004) can be used to efficiently compute the entire solution path (the solutions with all the possible cardinality on $\tilde{\mathbf{a}}$) of the problem in Eqn. (5).

One problem still remains. That is, the generalized eigenvector problem (3) is computationally expensive. In order to reduce the computational complexity, we have the following theorem:

Theorem 1 *Let **y** be the eigenvector of the following equation:*

$$L\boldsymbol{y} = \lambda D\boldsymbol{y} \tag{6}$$

If $X^T\boldsymbol{a} = \boldsymbol{y}$, then $\boldsymbol{a}$ is the eigenvector of the eigen-problem (3) with the same eigenvalue.

Proof We have $L\mathbf{y} = \lambda D\mathbf{y}$. At the left hand side of Eq. (3), replace $X^T\mathbf{a}$ by $\mathbf{y}$, thus we have

$$XLX^T\mathbf{a} = XL\mathbf{y} = X\lambda D\mathbf{y} = \lambda XD\mathbf{y} = \lambda XDX^T\mathbf{a}$$

Therefore, $\mathbf{a}$ is the eigenvector of eigen-problem (3) with the same eigenvalue λ.

Theorem (1) shows that instead of solving the eigen-problem in Eq. (3), the embedding result $\mathbf{y}$ can be directly obtained by solving Eq. (6). Since the graph is usually specially designed and sparse, the computation can be very efficient.

Once the embedding result $\mathbf{y}$ is obtained, we can apply lasso regression (Hastie, Tibshirani, & Friedman 2001) in Eqn. (5) to solve the optimization problem (1).

The Eigenvectors of Eigen-problem (6)

Noticing that $L = D - W$, we have

$$\begin{aligned} & L\mathbf{y} = \lambda D\mathbf{y} \\ \Rightarrow \quad & (D - W)\mathbf{y} = \lambda D\mathbf{y} \\ \Rightarrow \quad & W\mathbf{y} = (1-\lambda)D\mathbf{y} = \lambda' D\mathbf{y} \end{aligned} \tag{7}$$

Thus, finding the eigenvectors of the eigen-problem (6) with respect to the *smallest* eigenvalue is equivalent to finding the eigenvectors of eigen-problem (7) with respect to the *largest* eigenvalue.

Generally, we need to solve the eigen-problem in Eq. (7) to get the embedding vectors $\mathbf{y}$'s. Nevertheless, in some cases, *i.e.* LDA, the weight matrix W has a block diagonal structure and there is no need to solve the eigen-problem.

Without loss of generality, we assume that the data points in $\{\mathbf{x}_1, \cdots, \mathbf{x}_m\}$ are ordered according to their labels. Thus, W has a block-diagonal structure, as defined in Section 2. Since W is block-diagonal, its eigenvalues and eigenvectors[1] are the union of the eigenvalues and eigenvectors of its blocks (the latter padded appropriately with zeros) (Golub & Loan 1996). Let $W^{(t)}$ be the t-th diagonal block. That is, $W^{(t)}$ is a $m_t \times m_t$ matrix and $W^{(t)}_{ij} = \frac{1}{m_t}$, $\forall i, j$. It is straightforward to show that $W^{(t)}$ has eigenvector $\mathbf{e}^{(t)} \in \mathbb{R}^{m_t}$ associated with eigenvalue 1, where $\mathbf{e}^{(t)} = [1, 1, \cdots, 1]^T$. Also there is only one non-zero eigenvalue of $W^{(t)}$ because the rank of $W^{(t)}$ is 1. Thus, there are exactly c eigenvectors of W with the same eigenvalue 1. These eigenvectors are

$$\mathbf{y}_t = [\underbrace{0, \cdots, 0}_{\sum_{i=1}^{t-1} m_i}, \underbrace{1, \cdots, 1}_{m_t}, \underbrace{0, \cdots, 0}_{\sum_{i=t+1}^{c} m_i}]^T. \tag{8}$$

[1]It is easy to check that $D = I$ with the LDA W defined in Section 2. The generalized eigenvectors in Eqn. (7) reduce to ordinary eigenvectors of W.

Since 1 is a repeated eigenvalue of W, we could just pick any other c orthogonal vectors in the space spanned by $\{\mathbf{y}_k\}$, and define them to be our c eigenvectors. The vector of all ones $\mathbf{e}$ is naturally in the spanned space. This vector is useless since the responses of all the data points are the same. In reality, we can pick $\mathbf{e}$ as our first eigenvector and use Gram-Schmidt process to get the remaining $c-1$ orthogonal eigenvectors. The vector of all ones can then be removed.

In binary classification case, the above procedure will produce the following embedding vector

$$\mathbf{y} = [\underbrace{\frac{m}{m_1}, \cdots, \frac{m}{m_1}}_{m_1}, \underbrace{\frac{-m}{m_2}, \cdots, \frac{-m}{m_2}}_{m_2}]^T. \tag{9}$$

This is consistent with the previous well-known result on the relationship between LDA and regression for a binary problem (Hastie, Tibshirani, & Friedman 2001). The SPG algorithm proposed in this paper extends this relation to multi-class case. Moreover, our approach also establishes the connection between regression and many other graph based subspace learning algorithms, *e.g.*, Locality Preserving Projections (He & Niyogi 2003).

Computational Complexity of SPG

The SPG computation involves two steps: responses generation (i.e., calculate the eigenvectors of eigen-problem in Eq. (7)) and lasso regression.

Two of the most popular graphs are supervised block-diagonal graph (*e.g.*, LDA) and unsupervised p-nearest neighbor graph. For the weight matrix W of a block-diagonal graph, the cost of the first step is mainly the cost of Gram-Schmidt method, which is $O(mc^2)$ (Golub & Loan 1996). For a p-nearest neighbor graph, W is sparse and there are around p non-zero elements in each row of W. The Lanczos algorithm can be used to efficiently compute the first l eigenvectors of the eigen-problem in Eqn. (7) within $O(lqmp)$, where q is the number of iterations in Lanczos (Golub & Loan 1996).

By using the *Least Angel Regression* (LARS) algorithm, the entire solution path (the solutions with all the possible cardinality on $\tilde{\mathbf{a}}$) of lasso in Eqn. 5 can be computed in $O(n^3 + mn^2)$ (Efron *et al.* 2004). If we require $\mathrm{card}(\mathbf{a}) \leq k$, this cost can be reduced to $O(k^3 + mk^2)$ (Efron *et al.* 2004).

Considering $m \gg c$, SPG provides a sparse LDA solution with $O(n^3 + mn^2)$ complexity. Comparing to the $O(n^4 + mn^2)$ greedy algorithm described in (Moghaddam, Weiss, & Avidan 2006), SPG is much more efficient.

Experimental Results

In this section, we investigate the use of our algorithm for text clustering. The following five methods are compared in the experiment:

- **Baseline**: K-means on the original term-document matrix, which is treated as our baseline.
- **LSI**: K-means after Latent Semantic Indexing. LSI is essentially similar to PCA.

Table 1: Statistics of TDT2 and Reuters corpora.

	TDT2	Reuters
No. docs. used	9394	8067
No. clusters used	30	30
Max. cluster size	1844	3713
Min. cluster size	52	18
Med. cluster size	131	45
Avg. cluster size	313	269

Table 2: Statistics of clusters in TDT2 and Reuters corpora.

No. of clusters (c)	Avg. docs. #		Avg. terms #	
	TDT2	Reuters	TDT2	Reuters
2	605	641	6011	2486
3	939	1099	8342	3979
4	1180	1401	10102	5030
5	1660	1101	12773	4594
6	1650	1360	13042	5168
7	2255	1794	15423	6766
8	2557	2602	16761	7980
9	2725	2840	16943	8538
10	2987	2488	18842	8137

- **SPCA**: K-means after SparsePCA.
- **SPG**: K-means after SPG.
- **NMF**: Nonnegative Matrix Factorization-based clustering (Xu, Liu, & Gong 2003)). The weighted NMF-based clustering method is a recently proposed algorithm which has been shown to be very effective in document clustering (Xu, Liu, & Gong 2003).

Note that, our SPG algorithm needs to construct a graph over the documents. In this experiment, we set the parameter p (number of nearest neighbors) to 7.

All these algorithms are tested on the TDT2 corpus[2], and the Reuters-21578 corpus[3]. These two document corpora have been among the ideal test sets for document clustering purposes because documents in the corpora have been manually clustered based on their topics and each document has been assigned one or more labels indicating which topic/topics it belongs to.

The TDT2 corpus consists of data collected during the first half of 1998 and taken from 6 sources, including 2 newswires (APW, NYT), 2 radio programs (VOA, PRI) and 2 television programs (CNN, ABC). It consists of 11201 on-topic documents which are classified into 96 semantic categories. In this experiment, those documents appearing in two or more categories were removed, and only the largest 30 categories were kept, thus leaving us with 9,394 documents in total.

Reuters corpus contains 21578 documents which are grouped into 135 clusters. Compared with TDT2 corpus, the Reuters corpus is more difficult for clustering. In TDT2, the content of each cluster is narrowly defined, whereas in Reuters, documents in each cluster have a broader variety of content. Moreover, the Reuters corpus is much more unbalanced, with some large clusters more than 200 times larger than some small ones. In our test, we discarded documents with multiple category labels, and only selected the largest 30 categories. This left us with 8067 documents in total. Table 2 provides the statistics of the two document corpora.

Each document is represented as a term frequency (TF) vector and each vector is normalized to unit. For the purpose of reproducibility, we provide our algorithms and data sets used in these experiments at:
http://www.cs.uiuc.edu/homes/dengcai2/Data/data.html

Evaluation Metric The clustering result is evaluated by comparing the obtained label of each document with that provided by the document corpus. Two metrics, the accuracy (AC) and the normalized mutual information metric ($\overline{MI}$) are used to measure the clustering performance (Cai, He, & Han 2005), (Xu, Liu, & Gong 2003). Given a document $\mathbf{x}_i$, let r_i and s_i be the obtained cluster label and the label provided by the corpus, respectively. The AC is defined as follows:

$$AC = \frac{\sum_{i=1}^{n} \delta(s_i, map(r_i))}{n}$$

where n is the total number of documents and $\delta(x, y)$ is the delta function that equals one if $x = y$ and equals zero otherwise, and map(r_i) is the permutation mapping function that maps each cluster label r_i to the equivalent label from the data corpus. The best mapping can be found by using the Kuhn-Munkres algorithm (Lovasz & Plummer 1986).

Let C denote the set of clusters obtained from the ground truth and C' obtained from our algorithm. Their mutual information metric $MI(C, C')$ is defined as follows:

$$MI(C, C') = \sum_{c_i \in C, c'_j \in C'} p(c_i, c'_j) \cdot log_2 \frac{p(c_i, c'_j)}{p(c_i) \cdot p(c'_j)}$$

where $p(c_i)$ and $p(c'_j)$ are the probabilities that a document arbitrarily selected from the corpus belongs to the clusters c_i and c'_j, respectively, and $p(c_i, c'_j)$ is the joint probability that the arbitrarily selected document belongs to the clusters c_i as well as c'_j at the same time. In our experiments, we use the normalized mutual information $\overline{MI}$ as follows:

$$\overline{MI}(C, C') = \frac{MI(C, C')}{\max(H(C), H(C'))}$$

where $H(C)$ and $H(C')$ are the entropies of C and C', respectively. It is easy to check that $\overline{MI}(C, C')$ ranges from 0 to 1. $\overline{MI} = 1$ if the two sets of clusters are identical, and $\overline{MI} = 0$ if the two sets are independent.

Results The evaluations were also conducted with different number of clusters, ranging from 2 to 10. For each given cluster number c, 50 tests were conducted on different randomly chosen categories, and the average performance was computed over these 50 tests (except the 30 cluster case). For each test, K-means algorithm was applied 10

[2]Nist Topic Detection and Tracking corpus at http://www.nist.gov/speech/tests/tdt/tdt98/index.htm

[3]Reuters-21578 corpus is at http://www.daviddlewis.com/resources/testcollections/reuters21578/

Table 3: Clustering performance on TDT2

c	Accuracy (%)					Normalized Mutual Information (%)					Sparsity (%)	
	Baseline	LSI	SPCA	SPG	NMF	Baseline	LSI	SPCA	SPG	NMF	SPCA	SPG
2	97.7	98.7	99.2	**99.9**	99.2	91.3	94.7	96.6	**98.4**	96.6	98.3	98.5
3	88.4	91.0	94.3	**99.7**	95.7	81.5	84.3	88.1	**97.4**	90.7	99.4	99.3
4	85.7	87.4	90.8	**99.5**	92.4	82.0	82.3	85.2	**96.9**	87.7	99.5	99.3
5	82.4	81.7	85.1	**98.8**	92.2	79.0	77.7	79.9	**95.1**	86.6	99.8	99.5
6	79.0	79.0	83.3	**98.5**	88.0	78.1	78.0	80.4	**95.5**	84.2	99.3	99.3
7	74.5	72.4	76.7	**98.1**	83.1	74.5	73.2	75.0	**94.1**	79.9	99.3	99.2
8	70.1	68.1	71.8	**97.1**	79.7	71.5	69.4	71.2	**93.3**	76.2	99.4	99.3
9	72.3	70.6	73.6	**96.5**	84.8	75.1	73.9	74.6	**92.4**	81.8	99.6	99.7
10	69.2	66.7	71.0	**95.0**	81.5	73.1	71.3	72.6	**90.8**	78.3	99.8	99.6
Avg.	79.9	79.5	82.9	**98.1**	88.5	78.5	78.3	80.4	**94.9**	84.7	99.4	99.3

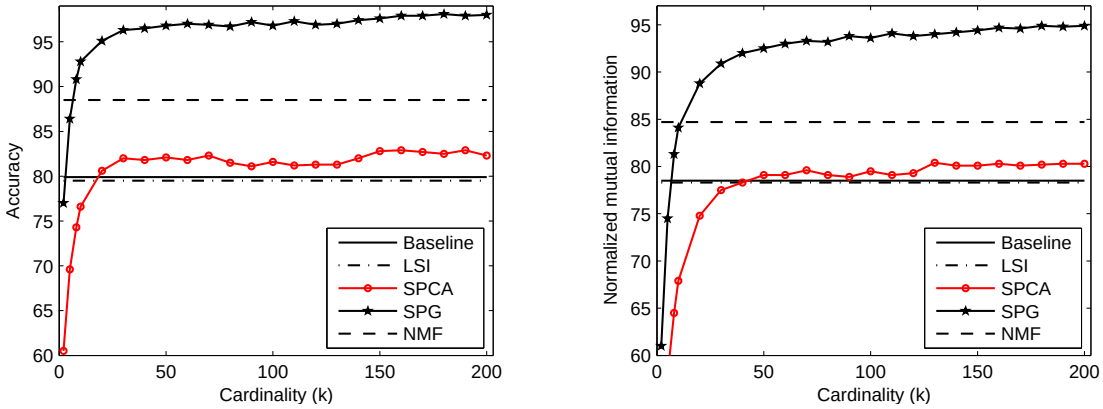

Figure 1: Accuracy and normalized mutual information vs. cardinality on TDT2 corpus

times with different start points and the best result in terms of the objective function of K-means was recorded. After LSI, SparsePCA, or SPG, how to determine the dimensionality of the subspace is still an open problem. In this experiment, we keep c dimensions for all the three algorithms as suggested by previous study (Cai, He, & Han 2005).

We show the performance change with the cardinality of basis functions in SparsePCA and SPG. As can be seen, the best performance is obtained with relatively small cardinality.

Conclusions

In this paper, we described a subspace learning algorithm called Sparse Projection over Graphs. We construct an affinity graph which can encode both discriminant and geometrical structure in the data. The sparse projections can be obtained by solving an optimization problem. We have also suggested an approach for solving this optimization problem by using techniques from spectral graph theory and lasso regression. Several experiments on document clustering were carried out on two databases. Our method was shown to outperform both PCA, LDA, and sparse PCA.

Several questions remains unclear and will be investigated in our future work. First, in this work, we use the simple nearest neighbor graph to encode both discriminating and geometrical structure of the data manifold. However, there is no reason to believe this is the only or the best way to construct the affinity graph. Second, our algorithm is linear, but it also can be performed in reproducing kernel Hilbert space (RKHS) which gives rise to nonlinear maps. The performance of SPG in RKHS needs to be further examined.

Acknowledgment

The work was supported in part by the U.S. National Science Foundation NSF IIS-05-13678, NSF BDI-05-15813 and MIAS (a DHS Institute of Discrete Science Center for Multimodal Information Access and Synthesis). Any opinions, findings, and conclusions or recommendations expressed here are those of the authors and do not necessarily reflect the views of the funding agencies.

References

Cai, D.; He, X.; and Han, J. 2005. Document clustering using locality preserving indexing. *IEEE Transactions on Knowledge and Data Engineering* 17(12):1624–1637.

Table 4: Clustering performance on Reuters

c	Accuracy (%)					Normalized Mutual Information (%)					Sparsity (%)	
	Baseline	LSI	SPCA	SPG	NMF	Baseline	LSI	SPCA	SPG	NMF	SPCA	SPG
2	83.2	83.9	83.3	**88.2**	84.0	49.0	**51.4**	48.3	49.6	45.9	98.8	99.2
3	73.8	73.6	75.3	**84.6**	76.7	48.1	**48.6**	48.5	48.3	46.3	95.7	95.5
4	66.6	67.7	67.9	**77.9**	71.2	47.1	48.0	47.6	**52.1**	47.2	97.8	96.0
5	62.4	61.7	62.5	**69.1**	67.5	48.1	48.7	49.3	**52.2**	49.6	96.3	95.9
6	60.2	63.2	62.0	62.4	**65.1**	50.2	**52.1**	51.2	49.8	49.4	96.1	97.5
7	52.3	53.4	53.6	**67.7**	59.0	44.3	44.7	44.5	**53.9**	44.2	97.0	97.0
8	47.0	48.4	47.9	**56.9**	54.0	41.1	41.4	40.3	**45.5**	39.3	98.0	97.7
9	42.1	42.5	42.7	**54.9**	51.8	36.9	36.7	36.2	**43.2**	36.8	97.9	97.7
10	45.6	44.4	44.8	**55.1**	52.3	42.3	41.2	41.0	**47.3**	40.7	97.9	97.5
Avg.	59.2	59.9	60.0	**68.5**	64.6	45.2	45.9	45.2	**49.1**	44.4	97.3	97.1

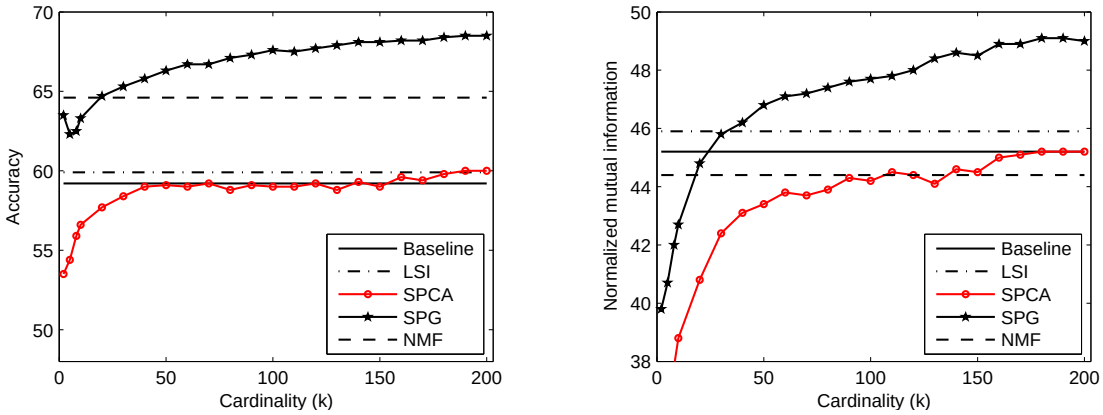

Figure 2: Accuracy and normalized mutual information vs. cardinality on Reuters corpus

Chung, F. R. K. 1997. *Spectral Graph Theory*, volume 92 of *Regional Conference Series in Mathematics*. AMS.

d'Aspremont, A.; Chaoui, L. E.; Jordan, M. I.; and Lanckriet, G. R. G. 2004. A direct formulation for sparse PCA using semidefinite programming. In *Advances in Neural Information Processing Systems 17*.

Duda, R. O.; Hart, P. E.; and Stork, D. G. 2000. *Pattern Classification*. Hoboken, NJ: Wiley-Interscience, 2nd edition.

Efron, B.; Hastie, T.; Johnstone, I.; and Tibshirani, R. 2004. Least angle regression. *Annals of Statistics* 32(2):407–499.

Fukunaga, K. 1990. *Introduction to Statistical Pattern Recognition*. Academic Press, 2nd edition.

Golub, G. H., and Loan, C. F. V. 1996. *Matrix computations*. Johns Hopkins University Press, 3rd edition.

Hastie, T.; Tibshirani, R.; and Friedman, J. 2001. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York: Springer-Verlag.

He, X., and Niyogi, P. 2003. Locality preserving projections. In *Advances in Neural Information Processing Systems 16*. Cambridge, MA: MIT Press.

He, X.; Yan, S.; Hu, Y.; Niyogi, P.; and Zhang, H.-J. 2005. Face recognition using laplacianfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27(3):328–340.

Lee, D. D., and Seung, H. S. 1999. Learning the parts of objects by non-negative matrix factorization. *Nature* 401:788–791.

Lovasz, L., and Plummer, M. 1986. *Matching Theory*. North Holland, Budapest: Akadémiai Kiadó.

Moghaddam, B.; Weiss, Y.; and Avidan, S. 2005. Spectral bounds for sparse PCA: Exact and greedy algorithms. In *Advances in Neural Information Processing Systems 18*.

Moghaddam, B.; Weiss, Y.; and Avidan, S. 2006. Generalized spectral bounds for sparse LDA. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, 641–648.

Xu, W.; Liu, X.; and Gong, Y. 2003. Document clustering based on non-negative matrix factorization. In *Proc. 2003 Int. Conf. on Research and Development in Information Retrieval (SIGIR'03)*, 267–273.

Zhou, H.; Hastie, T.; and Tibshirani, R. 2004. Sparse principle component analysis. Technical report, Statistics Department, Stanford University.