Locality Preserving Feature Learning

Quanquan Gu **Marina Danilevsky** **Zhenhui Li** **Jiawei Han**
Department of Computer Science
University of Illinois at Urbana-Champaign, IL 61801, USA
{qgu3,danilev1,zli28,hanj}@illinois.edu

Abstract

Locality Preserving Indexing (LPI) has been quite successful in tackling document analysis problems, such as clustering or classification. The approach relies on the *Locality Preserving Criterion*, which preserves the locality of the data points. However, LPI takes every word in a data corpus into account, even though many words may not be useful for document clustering. To overcome this problem, we propose an approach called Locality Preserving Feature Learning (LPFL), which incorporates feature selection into LPI. Specifically, we aim to find a subset of features, and learn a linear transformation to optimize the *Locality Preserving Criterion* based on these features. The resulting optimization problem is a mixed integer programming problem, which we relax into a constrained Frobenius norm minimization problem, and solve using a variation of Alternating Direction Method (ADM). ADM, which iteratively updates the linear transformation matrix, the residue matrix and the Lagrangian multiplier, is theoretically guaranteed to converge at the rate $O(\frac{1}{t})$. Experiments on benchmark document datasets show that our proposed method outperforms LPI, as well as other state-of-the-art document analysis approaches.

1 Introduction

Document representation plays an important role in solving text mining problems, such as document clustering and classification. The vector space model (VSM) [18] is one of the most popular models for document representation. In VSM, each document is represented as a bag of words. For a high dimensional vector space model, a dimensionality reduction phase is often applied first to reduce the size of the document representation. This avoids over-fitting, and the process of dimensionality reduction constructs a latent semantic representation of the document.

Latent Semantic Indexing (LSI) [8] was proposed as an approach for dimensionality reduction. LSI aims to find a subspace which best preserves the variance of the data, yielding an optimal representation in the sense that the global geometric structure of the documents is maintained. However, recent studies have shown that many real world datasets have a distribution which lies near a low-dimensional manifold, embedded in a high-dimensional ambient space [1] [23].

This characteristic of real world data serves as a motivation for Locality Preserving Indexing (LPI) [4]. LPI, originally proposed in [15], is based on the *Locality Preserving Criterion* [1], which says if two data points are close, their mappings should be close as well. LPI aims to find a subspace in which the local geometric structure of the document corpus is well preserved. It has been quite successful at document representation, and many variations of LPI have been proposed in the past decade [3] [5] [6]. However, one disadvantage of LPI is that it uses all of the discovered features to learn the projection. In reality, it is rare that every feature is useful for learning, and this is in fact a common problem among existing subspace learning methods [20]. Another method has been proposed in response to this problem, called Laplacian Score [14], which is a feature selection method based on *Locality Preserving Criterion.* Rather than seeking a particular subspace, Laplacian Score searches for a subset of features, which preserves the locality of the data. Although Laplacian Score can eliminate irrelevant and redundant features, it does not do any feature combination —that is done in LPI —which limits its performance.

Appearing in Proceedings of the 15^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2012, La Palma, Canary Islands. Volume XX of JMLR: W&CP XX. Copyright 2012 by the authors.

We can see that LPI suffers from a problem that is fixed in Laplacian score, while Laplacian score similarly suffers from a problem that does not affect LPI. Intuitively, Laplacian score and LPI can be thought of as complementary to each other, as each approach accounts for one issue, and ignores the other. Therefore, what we would like is an approach that integrates the feature selection of Laplacian score and the feature combination of LPI.

In this paper, we propose just such an approach, called *Locality Preserving Feature Learning* (LPFL). In particular, LPFL aims to simultaneously find a subset of features and a combination of features, to optimize the Locality Preserving Criterion. LPFL inherits the advantages of Laplacian score and LPI, as it is able to simultaneously discard the irrelevant features and transform the relevant ones. The resulting optimization problem is a mixed integer programming problem [2], which is difficult to solve. Therefore, we relax it into a constrained Frobenius norm minimization problem and solve it using a variation of Alternating Direction Method [21]. ADM iteratively updates the linear transformation matrix, the residue matrix and the Lagrangian multiplier. It is easy to implement and is theoretically guaranteed to converge.

The contributions of this paper are summarized as follows: (1) We propose a new feature learning approach based on Locality Preserving Criterion, which is able to achieve feature selection and transformation together; (2) the number of selected features is explicitly controlled in our method; (3) to solve the proposed model, we present a variation of ADM algorithm, which achieves the global solution of the proposed model with convergence rate $O(\frac{1}{t})$; and (4) experimental results on benchmark document data sets showed that the proposed method outperforms LPI, Laplacian score and other state-of-the-art related approaches.

The rest of this paper is organized as follows. In Section 2, we review LPI and Laplacian score. In Section 3, we present LPFL which incorporates feature selection into LPI. Experiments on benchmark document datasets are demonstrated in Section 5. We conclude the study and discuss future work in Section 6.

Notation We denote a dataset that consists of n data points as $\{\mathbf{x}_i\}_{i=1}^n$, where $\mathbf{x}_i \in \mathbb{R}^d$ denotes the i-th data point. The data matrix (e.g., term-document matrix) is denoted by $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^{d\times n}$, and the linear transformation matrix is denoted by $\mathbf{A} \in \mathbb{R}^{d\times l}$, mapping the input data into an l-dimensional subspace. Given a matrix $\mathbf{A} \in \mathbb{R}^{d\times l}$, we denote the i-th row of $\mathbf{A}$ by $\mathbf{a}^i$, and the j-th column of $\mathbf{A}$ by $\mathbf{a}_j$. The Frobenius norm of $\mathbf{A}$ is defined as $||\mathbf{A}||_F = \sqrt{\sum_i^d ||\mathbf{A}^i||_2^2} = \langle \mathbf{A}, \mathbf{A} \rangle = \mathrm{tr}(\mathbf{A}^T\mathbf{A})$ where $\langle \cdot, \cdot \rangle$ denotes the inner product of two matrices and $\mathrm{tr}(\cdot)$ denotes the trace of a matrix. The $L_{2,0}$-norm of $\mathbf{A}$ is defined as $||\mathbf{A}||_{2,0} = \mathrm{card}(||\mathbf{a}^1||_2, \ldots, ||\mathbf{a}^d||_2)$. $\mathbf{1}$ is a vector of all ones of some appropriate length, and $\mathbf{I}$ is the identity matrix of some appropriate size.

2 A Brief Review of Locality Preserving Criterion

In order to estimate and preserve the geometrical and topological properties of manifold data, *Locality Preserving Criterion* was proposed [1]. It assumes that if two data points $\mathbf{x}_i$ and $\mathbf{x}_j$ are close in the intrinsic geometry of the data distribution, then the mappings of this two points are also close to each other. Let $f(\mathbf{x}_i)$ be a function that produces the mapping of the original data point $\mathbf{x}_i$, we use $||f||^2_{\mathcal{M}}$ to measure the smoothness of f along the geodesics in the intrinsic geometry of the data. When we consider the case that the data is a compact submanifold $\mathcal{M} \subset \mathbb{R}^m$, a natural choice for $||f||^2_{\mathcal{M}}$ is

$$||f||^2_{\mathcal{M}} = \int_{\mathbf{x}\in\mathcal{M}} || \bigtriangledown_{\mathcal{M}} f||^2 d\mathbf{x} \tag{1}$$

where $\bigtriangledown_{\mathcal{M}} f$ is the gradient of f along the manifold $\mathcal{M}$. In reality, the data manifold $\mathcal{M}$ is unknown. Thus, $||f||^2_{\mathcal{M}}$ in Eq. (1) can not be computed. Recent studies on spectral graph theory [7] has demonstrated that $||f||^2_{\mathcal{M}}$ can be discretely approximated through a nearest neighbor graph on a scatter of data points. Given an affinity matrix $\mathbf{W} \in \mathbb{R}^{n\times n}$ of the graph, $||f||^2_{\mathcal{M}}$ is approximated as:

$$\begin{aligned} ||f||^2_{\mathcal{M}} &\approx \frac{1}{2}\sum_{ij} ||f_i - f_j||_2^2 W_{ij} \\ &= \mathrm{tr}(\mathbf{f}^T(\mathbf{D}-\mathbf{W})\mathbf{f}) \\ &= \mathrm{tr}(\mathbf{f}^T\mathbf{L}\mathbf{f}), \end{aligned} \tag{2}$$

where f_i is a shorthand for $f(\mathbf{x}_i)$, $\mathbf{f} = [f_1, \ldots, f_n]^T$, $\mathbf{D}$ is a diagonal matrix, called a degree matrix, with $D_{ii} = \sum_{j=1}^n W_{ij}$, and $\mathbf{L} = \mathbf{D} - \mathbf{W}$ is the graph Laplacian [7], which is a discrete approximation to the Laplace-Beltrami operator $\mathcal{M}$ on the manifold. Intuitively, the objective function incurs a heavy penalty if neighboring points $\mathbf{x}_i$ and $\mathbf{x}_j$ are mapped far apart.

For document representation, the affinity matrix is usually defined as:

$$W_{ij} = \begin{cases} \frac{\mathbf{x}_i^T\mathbf{x}_j}{||\mathbf{x}_i||_2||\mathbf{x}_j||_2}, & \text{if } \mathbf{x}_j \in \mathcal{N}(\mathbf{x}_i) \text{ or } \mathbf{x}_i \in \mathcal{N}(\mathbf{x}_j) \\ 0, & \text{otherwise,} \end{cases} \tag{3}$$

where $\frac{\mathbf{x}_i^T\mathbf{x}_j}{||\mathbf{x}_i||_2||\mathbf{x}_j||_2}$ is the cosine distance between $\mathbf{x}_i$ and $\mathbf{x}_j$, and $\mathcal{N}(\mathbf{x}_i)$ is the set of k-nearest neighbors of $\mathbf{x}_i$.

In the following, we will show that by different definitions of $f(\mathbf{x})$, Locality Preserving Indexing [15] [4] and Laplacian score [14] can be recovered.

2.1 Locality Preserving Indexing

In LPI [15] [4], the function $f(\mathbf{x})$ is defined as $f(\mathbf{x}) = \mathbf{A}^T\mathbf{x}$ where $\mathbf{A} \in \mathbb{R}^{d\times l}$ is a linear transformation matrix. Submit $f(\mathbf{x})$ back into Eq. (2), we obtain the objective function of LPI as follows:

$$\begin{aligned} \arg\min_{\mathbf{A}} \quad & \mathrm{tr}(\mathbf{A}^T\mathbf{X}\mathbf{L}\mathbf{X}^T\mathbf{A}) \\ \text{s.t.} \quad & \mathbf{A}^T\mathbf{X}\mathbf{D}\mathbf{X}^T\mathbf{A} = \mathbf{I}, \end{aligned} \tag{4}$$

where $\mathbf{A}^T\mathbf{X}\mathbf{D}\mathbf{X}^T\mathbf{A} = \mathbf{I}$ is added to avoid the trivial solution. It is easy to show that the problem in Eq. (4) is equivalent to the following *Ratio Trace* problem [9]:

$$\arg\min_{\mathbf{A}} \mathrm{tr}\left((\mathbf{A}^T\mathbf{X}\mathbf{L}\mathbf{X}^T\mathbf{A})(\mathbf{A}^T\mathbf{X}\mathbf{D}\mathbf{X}^T\mathbf{A})^{-1}\right). \tag{5}$$

2.2 Laplacian Score

In Laplacian score, we define $f(\mathbf{x})$ as $f(\mathbf{x}) = \mathbf{p} \odot \mathbf{x}$, where $\mathbf{p} = (p_1, \ldots, p_d)^T$ and $p_i \in \{0,1\}, i = 1, \ldots, d$, to represent whether a feature is selected or not, and $\odot$ is element-wise product. Then feature selection based on Laplacian score can be written as:

$$\begin{aligned} \arg\min_{\mathbf{p}} \quad & \mathrm{tr}\{(\mathrm{diag}(\mathbf{p})\mathbf{X}\mathbf{L}\mathbf{X}^T\mathrm{diag}(\mathbf{p})) \\ & (\mathrm{diag}(\mathbf{p})\mathbf{X}\mathbf{D}\mathbf{X}^T\mathrm{diag}(\mathbf{p}))^{-1}\}, \\ \text{s.t.} \quad & \mathbf{p} \in \{0,1\}^d, \mathbf{p}^T\mathbf{1} = m, \end{aligned} \tag{6}$$

where $\mathrm{diag}(\mathbf{p})$ is a diagonal matrix whose diagonal elements are p_i's. In order to indicate that m features are selected, we constrain $\mathbf{p}$ by $\mathbf{p}^T\mathbf{1} = m$. There are $\binom{d}{m}$ candidate feature subsets, hence Eq. (6) is a combinatorial optimization problem. Like other feature selection approaches [13], [14] uses a greedy algorithm and considers each feature individually. In particular, the Laplacian score of the j-th feature is defined as:

$$LS_j = \frac{\mathbf{x}^j\mathbf{L}(\mathbf{x}^j)^T}{\mathbf{x}^j\mathbf{D}(\mathbf{x}^j)^T}, \tag{7}$$

where $\mathbf{x}^j$ is the jth row of $\mathbf{X}$. The greedy algorithm is to compute the Laplacian score for each feature, and then select the top-m features with the lowest scores. The selected features as a whole are suboptimal.

3 The Proposed Method

Our proposed method incorporates feature selection into Locality Preserving Indexing. The key idea is to find a subset of features, based on which we learn a linear transformation under the Locality Preserving Criterion. We define $f(\mathbf{x}) = \mathbf{A}^T(\mathbf{p} \odot \mathbf{x})$ where $\mathbf{A}$ and $\mathbf{p}$ are defined as before. By substituting $f(\mathbf{x})$ into Eq. (2), we get:

$$\begin{aligned} \arg\min_{\mathbf{A},\mathbf{p}} \quad & \mathrm{tr}(\mathbf{A}^T\mathrm{diag}(\mathbf{p})\mathbf{X}\mathbf{L}\mathbf{X}^T\mathrm{diag}(\mathbf{p})\mathbf{A}) \\ \text{s.t.} \quad & \mathbf{A}^T\mathrm{diag}(\mathbf{p})\mathbf{X}\mathbf{D}\mathbf{X}^T\mathrm{diag}(\mathbf{p})\mathbf{A} = \mathbf{I} \\ & \mathbf{p} \in \{0,1\}^d, \mathbf{p}^T\mathbf{1} = m, \end{aligned} \tag{8}$$

which is a mixed integer programming [2]. We refer to Eq. (8) as the *Locality Preserving Feature Learning*(LPFL) approach, because it is able to simultaneously do feature selection and subspace learning, inheriting the advantages of Laplacian score and LPI. LPFL finds a subset of useful features, based on which it generates new features by feature transformation. Setting $\mathbf{p} = \mathbf{1}$, Eq. (8) reduces to LPI as in Eq. (5). Similarly, setting $\mathbf{A} = \mathbf{I}$, Eq. (8) degenerates to Laplacian score as in Eq.(6). Hence both LPI and Laplacian score can be seen as special cases of LPFL. In addition, the objective functions corresponding to LPI and Laplacian score are upper bounds of the objective function of LPFL.

The formulation of Eq. (8) is difficult to solve, so we will now reformulate the problem:

Theorem 1. *Let $\mathbf{Y} \in \mathbb{R}^{n\times m}$ be a matrix where each column is an eigenvector of eigen-problem $\mathbf{L}\mathbf{y} = \lambda\mathbf{D}\mathbf{y}$. If there exists a matrix $\mathbf{A} \in \mathbb{R}^{d\times m}$ and $\mathbf{p}$ where $\mathbf{p} \in \{0,1\}^d, \mathbf{p}^T\mathbf{1} = m$ such that $\mathbf{X}^T diag(\mathbf{p})\mathbf{A} = \mathbf{Y}$, then each column of $\mathbf{A}$ is an eigenvector of eigen-problem $diag(\mathbf{p})\mathbf{X}\mathbf{L}\mathbf{X}^T diag(\mathbf{p})\mathbf{a} = \lambda diag(\mathbf{p})\mathbf{X}\mathbf{D}\mathbf{X}^T diag(\mathbf{p})\mathbf{a}$ with the same eigenvalue λ.*

Proof. The proof can be found in the supplemental material. □

According to Theorem 1, the optimal $(\mathbf{A}^*, \mathbf{p}^*)$ that minimizes the problem in Eq. (8) can be obtained in the following two steps: (1) Solve the eigen-problem $\mathbf{L}\mathbf{Y} = \mathbf{\Lambda}\mathbf{D}\mathbf{Y}$ to get $\mathbf{Y}$; and (2) find $\mathbf{A}$ and $\mathbf{p}$ which satisfy $\mathbf{X}^T\mathrm{diag}(\mathbf{p})\mathbf{A} = \mathbf{Y}$.

Finding a solution $(\mathbf{A}, \mathbf{p})$ such that $\mathbf{X}^T\mathrm{diag}(\mathbf{p})\mathbf{A} = \mathbf{Y}$ is usually impossible. Hence, we introduce a residue matrix $\mathbf{E}$, and solve the following problem instead:

$$\begin{aligned} \min_{\mathbf{A},\mathbf{E},\mathbf{p}} \quad & \frac{1}{2}||\mathbf{E}||_F^2, \\ \text{s.t.} \quad & \mathbf{X}^T\mathrm{diag}(\mathbf{p})\mathbf{A} + \mathbf{E} = \mathbf{Y} \\ & \mathbf{p} \in \{0,1\}^d, \mathbf{p}^T\mathbf{1} = m, \end{aligned} \tag{9}$$

where $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_c] \in \mathbb{R}^{n\times c}$, and $\mathbf{y}_k$ is the eigenvector obtained in Step (1). In the ideal case, the optimal $\mathbf{E}$ is a zero matrix, and $\mathbf{X}^T\mathrm{diag}(\mathbf{p})\mathbf{A} = \mathbf{Y}$ holds exactly. In reality, we can obtain an $\mathbf{E}$ which is close to a zero matrix. In other words, $||\mathbf{E}||_F$ is close to zero. However, Eq. (9) is still a mixed integer programming [2], and therefore still difficult to solve.

Assume the optimal solutions of Eq. (9) are $(\mathbf{A}^*, \mathbf{p}^*)$. Then, since $\mathbf{p}^*$ is a binary vector, $\text{diag}(\mathbf{p}^*)\mathbf{A}^*$ is a matrix where the elements of many rows are all zeros. We, therefore, absorb the indicator variable $\mathbf{p}$ into $\mathbf{A}$, and use $L_{2,0}$-norm on $\mathbf{A}$ to achieve feature selection, resulting in the following problem:

$$\begin{aligned} \arg\min_{\mathbf{A},\mathbf{E}} \quad & \frac{1}{2}||\mathbf{E}||_F^2, \\ \text{s.t.} \quad & \mathbf{X}^T\mathbf{A} + \mathbf{E} = \mathbf{Y}, ||\mathbf{A}||_{2,0} \leq m. \end{aligned} \tag{10}$$

Eq. (10) is a constrained Frobenius norm minimization problem. Note that although the feasible region defined by $||\mathbf{A}||_{2,0} \leq m$ is not convex, we later will show that the global optimal solution can still be achieved. In the following section, we present an algorithm for solving Eq. (10). Due to the equality constraint, the most natural approach for solving the problem in Eq. (10) is an augmented Lagrangian multiplier method [2]. We derive an algorithm based on a variation of Alternating Direction Method (ADM) [21] for solving Eq. (10), which is an approximate augmented Lagrangian multiplier method. It is worth noting that ADM has been successfully applied for the recovery of single sparse vectors [21] and jointly sparse vectors [16].

3.1 Alternating Direction Method

The standard ADM was designed to solve the following structured optimization problem:

$$\begin{aligned} \min_{\mathbf{x},\mathbf{y}} \quad & f(\mathbf{x}) + g(\mathbf{y}) \\ \text{s.t.} \quad & \mathbf{P}\mathbf{x} + \mathbf{Q}\mathbf{y} = \mathbf{b}, \end{aligned} \tag{11}$$

where $\mathbf{x}$ and $\mathbf{y}$ are vectors, f and g are two real-valued functions, and $\mathbf{P}, \mathbf{Q}, \mathbf{b}$ are matrices and a vector of appropriate dimensions. Variables $\mathbf{x}$ and $\mathbf{y}$ are separate in the objective function and coupled only in the constraint. The augmented Lagrangian function of Eq. (11) is

$$\begin{aligned} L(\mathbf{x},\mathbf{y},\boldsymbol{\lambda},\mu) &= f(\mathbf{x}) + g(\mathbf{y}) + \boldsymbol{\lambda}^T(\mathbf{P}\mathbf{x} + \mathbf{Q}\mathbf{y} - \mathbf{b}) \\ &+ \frac{\mu}{2}||\mathbf{P}\mathbf{x} + \mathbf{Q}\mathbf{y} - \mathbf{b}||_F^2, \end{aligned} \tag{12}$$

where $\boldsymbol{\lambda}$ is a Lagrangian multiplier, and μ is a positive scalar. ADM employs the separability structure in Eq. (11) and replaces the joint optimization with respect to $(\mathbf{x}, \mathbf{y})$ by two simpler subproblems. The algorithm of ADM is outlined in Algorithm 1.

The convergence result of Algorithm 1 has been established when $\rho \in (0, \frac{\sqrt{5}+1}{2})$ in [10].

Algorithm 1 Alternating Direction Method

```
Initialize: $\rho, \mu, \lambda = 0, \mathbf{x} = 0, \mathbf{y} = 0$;
repeat
  Solve $\mathbf{x}_{t+1} = \arg\min_{\mathbf{x}} L(\mathbf{x}, \mathbf{y}_t, \boldsymbol{\lambda}_t, \mu_t)$
  Solve $\mathbf{y}_{t+1} = \arg\min_{\mathbf{y}} L(\mathbf{x}_t, \mathbf{y}, \boldsymbol{\lambda}_t, \mu_t)$
  Update $\boldsymbol{\lambda}_{t+1} = \boldsymbol{\lambda}_t + \mu_t(\mathbf{P}\mathbf{x}_{t+1} + \mathbf{Q}\mathbf{y}_{t+1} - \mathbf{b})$
  Update $\mu_{t+1} = \rho\mu_t$
until convergence
```

Since the problem in Eq. (10) has the same structure as that in Eq. (11) except the additional inequality constraint $||\mathbf{A}||_{2,0} \leq m$, we solve it by a variation of ADM. The augmented Lagrangian function of Eq. (10) is:

$$\begin{aligned} L(\mathbf{A},\mathbf{E},\boldsymbol{\Lambda}) &= \frac{1}{2}||\mathbf{E}||_F^2 - \langle \boldsymbol{\Lambda}, \mathbf{X}^T\mathbf{A} + \mathbf{E} - \mathbf{Y} \rangle \\ &+ \frac{\mu}{2}||\mathbf{X}^T\mathbf{A} + \mathbf{E} - \mathbf{Y}||_F^2, \end{aligned} \tag{13}$$

where $\boldsymbol{\Lambda}$ is a Lagrangian multiplier, and $\mu > 0$ is a penalty parameter.

We will now derive the optimization algorithm for the two subproblems with respect to $\mathbf{E}$ and $\mathbf{A}$, and the update rule for $\boldsymbol{\Lambda}$ based on ADM as introduced above.

3.1.1 Solving E

Given $\mathbf{A}$ and $\boldsymbol{\Lambda}$, we solve the first subproblem with respect to $\mathbf{E}$. By removing terms that do not depend on $\mathbf{E}$, and adding proper terms that do not depend on $\mathbf{E}$, the optimization problem with respect to $\mathbf{E}$ reduces to:

$$\begin{aligned} \min_{\mathbf{E}} L(\mathbf{E}) &= \frac{1}{2}||\mathbf{E}||_F^2 - \langle \boldsymbol{\Lambda}, \mathbf{X}^T\mathbf{A} + \mathbf{E} - \mathbf{Y} \rangle \\ &+ \frac{\mu}{2}(||\mathbf{X}^T\mathbf{A} + \mathbf{E} - \mathbf{Y}||_F^2). \end{aligned} \tag{14}$$

Taking the gradient of $L(\mathbf{E})$ with respect to $\mathbf{E}$ and setting it to zero, we obtain:

$$\frac{\partial L(\mathbf{E})}{\partial \mathbf{E}} = \mathbf{E} - \boldsymbol{\Lambda} - \mu\mathbf{Y} + \mu\mathbf{X}^T\mathbf{A} + \mu\mathbf{E} = 0, \tag{15}$$

which leads to the closed-form solution of $\mathbf{E}$:

$$\mathbf{E} = \frac{1}{1+\mu}(\boldsymbol{\Lambda} + \mu\mathbf{Y} - \mu\mathbf{X}^T\mathbf{A}). \tag{16}$$

3.1.2 Solving A

Next, given $\mathbf{E}$ and $\boldsymbol{\Lambda}$, we solve the second subproblem with respect to $\mathbf{A}$. By removing the terms that do not depend on $\mathbf{A}$ and adding proper terms that do not depend on $\mathbf{A}$, the optimization problem with respect to $\mathbf{A}$ reduces to

$$\begin{aligned} \min_{\mathbf{A}} \quad & L(\mathbf{A}) = \frac{\mu}{2}||\mathbf{X}^T\mathbf{A} + \mathbf{E} - \mathbf{Y} - \frac{1}{\mu}\boldsymbol{\Lambda})||_F^2 \\ \text{s.t.} \quad & ||\mathbf{A}||_{2,0} \leq m. \end{aligned} \tag{17}$$

As in [21] [16], we approximate the second term of the objective function by its second order Taylor expansion at $\mathbf{A}_t$:

$$\begin{aligned} & ||\mathbf{X}^T\mathbf{A} + \mathbf{E} - \mathbf{Y} - \frac{1}{\mu}\mathbf{\Lambda})||_F^2 \\ = \ & ||\mathbf{X}^T\mathbf{A}_t + \mathbf{E} - \mathbf{Y} - \frac{1}{\mu}\mathbf{\Lambda})||_F^2 \\ + \ & 2\langle \mathbf{X}(\mathbf{X}^T\mathbf{A} + \mathbf{E} - \mathbf{Y} - \frac{1}{\mu}\mathbf{\Lambda}), \mathbf{A} - \mathbf{A}_t\rangle \\ + \ & \frac{1}{\tau}||\mathbf{A} - \mathbf{A}_t||_F^2. \end{aligned} \tag{18}$$

Substituting Eq. (18) into Eq. (17), we obtain:

$$\begin{aligned} L(\mathbf{A}) = \ & \frac{\mu}{2}\{2\langle(\mathbf{X}(\mathbf{X}^T\mathbf{A} + \mathbf{E} - \mathbf{Y} - \frac{1}{\mu}\mathbf{\Lambda}), \mathbf{A} - \mathbf{A}_t\rangle \\ + \ & \frac{1}{\tau}||\mathbf{A} - \mathbf{A}_t||_F^2\}. \end{aligned} \tag{19}$$

For the sake of simplicity, we denote $\mathbf{B}_t = \mathbf{X}(\mathbf{X}^T\mathbf{A}_t + \mathbf{E} - \mathbf{Y} - \frac{1}{\mu}\mathbf{\Lambda})$ and $\mathbf{C}_t = \mathbf{A}_t - \tau\mathbf{B}_t$. The minimization of Eq. (19) then takes the following form:

$$\begin{aligned} \mathbf{A}_{t+1} = \ & \arg\min_{\mathbf{A}} \frac{\mu}{\tau}||\mathbf{A} - \mathbf{C}_t||_F^2, \\ & \text{s.t.}||\mathbf{A}||_{2,0} \leq m. \end{aligned} \tag{20}$$

Although the feasible region defined by $||\mathbf{A}||_{2,1} \leq m$ is non-convex, the global optimal solution of the above problem can be obtained by the following theorem.

Theorem 2. *The global optimal solution of Eq. (20) is*

$$\mathbf{a}^{\pi(i)*} = \begin{cases} \mathbf{c}_t^{\pi(i)}, & i \leq m \\ \mathbf{0}, & otherwise. \end{cases} \tag{21}$$

where $\pi(i)$ is a sorting function such that $||\mathbf{c}_t^{\pi(1)}|| \geq ||\mathbf{c}_t^{\pi(2)}|| \geq \ldots, \geq ||\mathbf{c}_t^{\pi(d)}||$.

3.1.3 Updating $\mathbf{\Lambda}$

Given $\mathbf{A}$ and $\mathbf{E}$, the optimization problem with respect to $\mathbf{\Lambda}$ is reduced to

$$L(\mathbf{\Lambda}) = \langle \mathbf{\Lambda}, \mathbf{Y} - \mathbf{X}^T\mathbf{A} - \mathbf{E}\rangle. \tag{22}$$

The gradient of $L(\mathbf{\Lambda})$ is

$$\frac{\partial L(\mathbf{\Lambda})}{\partial \mathbf{\Lambda}} = \mathbf{Y} - \mathbf{X}^T\mathbf{A} - \mathbf{E}. \tag{23}$$

According to the augmented Lagrangian method [2], $\mathbf{\Lambda}$ is updated as

$$\mathbf{\Lambda} = \mathbf{\Lambda} + \mu(\mathbf{Y} - \mathbf{X}^T\mathbf{A} - \mathbf{E}), \tag{24}$$

where μ is updated as $\mu = \rho\mu$ and $\rho > 0$ is a scalar.

We summarize LPFL in Algorithm 2.

Algorithm 2 Locality Preserving Feature Learning

```
Initialize: $m, \mathbf{A} = 0, \mathbf{E} = 0, \rho = 1.5, \mu = 1e - 4$;
Compute $\mathbf{Y}$ by eigen-decomposition $\mathbf{LY} = \mathbf{\Lambda DY}$.
repeat
    Compute $\mathbf{E}_{t+1} = \frac{1}{1+\mu}(\mathbf{\Lambda}_t + \mu_t\mathbf{Y} - \mu_t\mathbf{X}^T\mathbf{A}_t)$
    Compute $\mathbf{B}_{t+1} = \mathbf{X}(\mathbf{X}^T\mathbf{A}_t + \mathbf{E}_{t+1} - \mathbf{Y} - \frac{1}{\mu}_t\mathbf{\Lambda}_t)$
    Solve $\mathbf{A}_{t+1}$ by $\arg\min_{\mathbf{A}} \frac{1}{2}||\mathbf{A} - (\mathbf{A}_t - \tau\mathbf{B}_{t+1})||_F^2, \text{s.t.}||\mathbf{A}||_{2,0} \leq m$
    Update $\mathbf{\Lambda}_{t+1} = \mathbf{\Lambda}_t + \mu_t(\mathbf{Y} - \mathbf{X}^T\mathbf{A}_{t+1} - \mathbf{E}_{t+1})$
    Update $\mu_{t+1} = \rho\mu_t$
until convergence
```

3.2 Theoretical Analysis

The proposed method has several nice properties.

First, the optimal objective function can be bounded by the generalized eigenvalues of the full matrices $\mathbf{L}'$ and $\mathbf{D}'$.

Theorem 3. *Let $\mathbf{L}' = \mathbf{XLX}^T$, $\mathbf{D}' = \mathbf{XDX}^T$, and $\lambda_i(\mathbf{L}', \mathbf{D}'), i = 1, \ldots, d$ be the generalized value of $\mathbf{L}'$ and $\mathbf{D}'$ sorted in ascending order. The optimal objective function value J of LPFL in Eq. (8) is bounded by*

$$\sum_{i=1}^{l} \lambda_i(\mathbf{L}', \mathbf{D}') \leq J \leq \sum_{i=1}^{l} \lambda_{i+d-m}(\mathbf{L}', \mathbf{D}').$$

where l is the dimension of the subspace learned by $\mathbf{A}$, and m is the number of selected features.

This provides a possible way to do model selection. That is, to select an appropriate m before running Algorithm 2.

Second, the derived variation of ADM is guaranteed to converge to the global optimal solution.

Theorem 4. *For Algorithm 2, if $\sum_{t=1}^{+\infty} \mu_t^{-2}\mu_{t+1} < +\infty$, then $(\mathbf{A}_t, \mathbf{E}_t, \mathbf{\Lambda}_t)$ converges to an optimal solution $(\mathbf{A}^*, \mathbf{E}^*, \mathbf{\Lambda}^*)$.*

Actually, we can prove that the convergence rate of Algorithm 2 is $O(\frac{1}{t})$.

Theorem 5. *Let $\boldsymbol{\theta} = [vec(\mathbf{A})^T, vec(\mathbf{E})^T, vec(\mathbf{\Lambda})^T]^T$ where $vec(\mathbf{A})$ denotes the vectorization of matrix $\mathbf{A}$, $\mathbf{H} = diag([\mathbf{I} - \mu\mathbf{XX}^T, \mu\mathbf{I}, \frac{1}{\mu}\mathbf{I}])$ is a block diagonal matrix where $\mathbf{I}$ is the identity matrix with appropriate size, and $F(\boldsymbol{\theta}) = [-vec(\mathbf{X\Lambda})^T, vec(\mathbf{\Lambda})^T, vec(\mathbf{X}^T\mathbf{A} + \mathbf{E} - \mathbf{Y})^T]^T$, Let $\{\boldsymbol{\theta}_t\}$ be the sequence generated by Algorithm 2. For any integer number $t > 0$, let $\bar{\boldsymbol{\theta}}_t$ be defined by $\bar{\boldsymbol{\theta}}_t = \frac{1}{t+1}\sum_{k=0}^{t} \boldsymbol{\theta}_t$, Then, we have*

$$\frac{1}{2}||\bar{\mathbf{E}}_t||_F^2 - \frac{1}{2}||\mathbf{E}^*||_F^2 + (\bar{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^*)^T F(\boldsymbol{\theta}^*) \leq \frac{1}{2(t+1)}||\boldsymbol{\theta}_0 - \boldsymbol{\theta}^*||_H^2$$

where $||\boldsymbol{\theta}_0 - \boldsymbol{\theta}^*||_H^2 = (\boldsymbol{\theta}_0 - \boldsymbol{\theta}^*)^T\mathbf{H}(\boldsymbol{\theta}_0 - \boldsymbol{\theta}^*)$

4 Related Work

In this section, we review several related works to our proposed method.

For better interpretability, [5] proposed sparse locality preserving indexing (SLPI), which is based on ℓ_1-norm regularization on each column of the linear transformation matrix, i.e., $\mathbf{a}_i$. Due to the nature of the ℓ_1 penalty, some entries in $\mathbf{a}_i$ will be shrunk to exact zero. However, SLPI does not lead to feature selection, because each column of the linear transformation matrix is optimized individually, and their sparsity patterns are independent. Based on the above observation, [11] proposed a joint feature selection and subspace learning (FSSL) approach. Rather than using ℓ_1-norm regularization, they use $L_{2,1}$-norm regularization on the linear transformation matrix. The resultant model is approximately solved by a two-stage algorithm. The algorithm is proved to converge. However, its convergence rate remains unclear. In contrast, the convergence rate of the proposed optimization algorithm is strictly given.

Incorporating feature selection into transformation also received attention in supervised dimensionality reduction. For example, [17] proposed linear discriminant feature selection (LDFS), which modifies Linear Discriminant Analysis to admit feature selection. Their optimization algorithm is based on Newton method, which is very time consuming. [12] proposed linear discriminant dimensionality reduction (LDDR), which is based on an equivalent linear regression formulation of LDA [22]. However, such kind of equivalence only holds under a very strict condition, which limits its application in general. Though our proposed method is an unsupervised method for data representation, the technique proposed in this paper can actually be adapted to solve those issues of the supervised approaches mentioned above.

It is also worth noting that the number of selected features is explicitly controlled by m in the proposed method, while the number of features is implicitly controlled by a regularization parameter in the other methods mentioned before [5] [11] [17] [12].

5 Experiments

In this section, we evaluate our proposed method and compare it with other state-of-the-art document representation methods: Latent Semantic Indexing (LSI) [8] and Locality Preserving Indexing (LPI) [4]. In addition, we compare LPFL with sparse LPI (SLPI) [5], which seeks a sparse projection. We also compare LPFL with Laplacian score (LS) [14]. In addition, we compare the proposed method with unsupervised version of joint feature selection and subspace learning (FSSL) method [11]. We also investigate the two-phase approach of Laplacian score followed by LPI (LS+LPI), which is the most intuitive way to integrate feature selection with LPI. We use K-means clustering as the baseline method. All of the experiments were performed in Matlab on an Intel Core2 Duo 2.8GHz Windows 7 machine with 4GB memory.

5.1 Datasets and Evaluation Measures

We use two text datasets that are used in [4][1].

TDT2 consists of data collected during the first half of 1998 and taken from 6 sources, including 2 newswires (APW, NYT), 2 radio programs (VOA, RPI) and 2 television programs (CNN, ABC). It consists of 11201 documents which are classified into 96 semantic categories. In this experiment, documents appearing in two or more categories were removed, and only the largest 30 categories were kept, leaving us with 9394 documents.

Reuters contains 21578 documents which are grouped into 135 clusters. The Reuters corpus is much more unbalanced, as some large clusters are more than 200 times larger than some small clusters. In our experiment, we discarded documents with multiple category labels, and only selected the largest 30 categories, leaving us with 8067 documents.

To evaluate the clustering results, we adopt the performance measures used in [19]: clustering accuracy and normalized mutual information. They are the standard measures widely used for clustering.

5.2 Parameter Settings

The clustering evaluations were conducted with varying numbers of clusters, ranging from 2 to 10. For example, when $c = 2$, we randomly choose 2 classes from the total 30 classes in the whole text dataset to do clustering. For each given cluster number c, 20 tests were conducted on different randomly chosen categories, and the average performance was computed over these 20 tests. For each test, the K-means algorithm was applied 10 times with different start points and the best result in terms of the objective function of K-means was recorded.

Note that, LPI, SPLI, Laplacian score, FSSL, and LPFL need to construct a graph on the documents. In this experiment, we use the same graph for all these methods. In particular, we use the affinity matrix from Eq. (3), where k is set to 5. K-means is applied after the dimensionality reduction to do clustering. In

[1] http://www.zjucadcg.cn/dengcai/Data/TextData.html

general, the clustering results after those dimensionality reduction methods vary with the dimensions of the subspace, or the number of selected features. In this experiment, for those subspace learning methods, we keep c dimensions as suggested by previous work [4]. For Laplacian score, the number of features is tuned by searching the grid $\{1000, 1500, 2000, \dots, 5000\}$. Given the above parameter settings, for LPI and Laplacian score, there are no parameters to be set. For the two-phase LS+LPI approach, we first use Laplacian Score to select features and then we perform LPI to reduce the dimensionality. The number of features selected by Laplacian Score is tuned the same as above. For SLPI, we tune the parameter β in [5] by searching the grid $\{10, 20, \dots, 100\}$ according to [5]. For FSSL, we use the same parameters as in their paper [11]. For LPFL, the number of features m is tuned by the same strategy as in Laplacian score.

5.3 Convergence

Before reporting the clustering results, we first examine the rate of convergence of ADM in Algorithm 2. In Figure 1, we plot the objective function value in Eq. (10) versus the number of iterations on the TDT2 dataset with $c = 2$. In the figure, the y-axis is the value of the objective function and the x-axis denotes the iteration number.

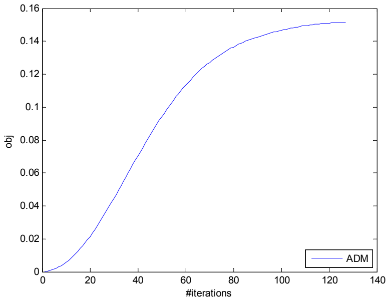
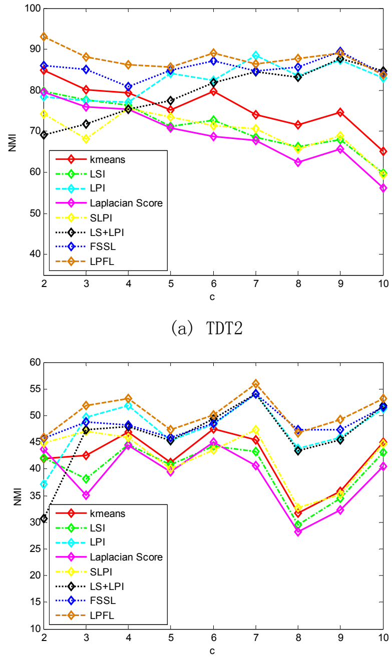


Figure 1: The objective function value of LPFL with respect to the number of iterations for ADM on the TDT2 dataset with $c = 2, m = 2000$.

We can see that ADM converges within roughly 130 iterations. This is consistent with the theoretical result presented in Section 3. It may be surprised to see that the objective function is increasing rather than decreasing, although the optimization problem in Eq. (10) is a minimization problem. This is due to the nature of the augmented Lagrangian method. When the penalty scalar μ is small, the equality constraint does not strictly hold and the feasible solution space is large, so we can achieve a small objective function value. Once the penalty scalar μ increases, the feasible solution space becomes smaller, and a larger objective function value will be achieved.

5.4 Clustering Results.

The clustering accuracy on the TDT2 and Reuters21578 datasets are shown in Table 1 and Table 2 respectively. To save space, the normalized mutual information on the TDT2 and Reuters21578 datasets are shown in Figure 2 in a compact way. For each c, we did paired t-tests at 95% significance level between the proposed method and the other methods. If it is significant over all the other methods, the corresponding entry of LPFL is bolded. Otherwise, the entry is not bolded.



(a) TDT2

(b) Reuters21578

Figure 2: Normalized mutual information while varying the number of clusters on (a) TDT2 and (b) Reuters21578 datasets. The graphs are best viewed in color.

A number of interesting results may be observed:

- On the TDT2 dataset, when the number of clusters is small, e.g., less than 3, LPI is comparable to K-means and LSI. However, as the number of clusters increases, e.g., larger than 4, LPI significantly outperforms K-means and LSI. This is because as the number of clusters increases, the document distribution becomes more like a manifold. In this case, the local geometric structure is more crucial for better clustering than the global geometric structure. On the Reuters21578 dataset, LPI outperforms K-means and LSI. As mentioned previously, the Reuters21578 corpus is more difficult to cluster due to its unbalanced clusters. By

Table 1: Clustering accuracy on the TDT2 dataset

c	Baseline	LSI	LPI	LS	SLPI	LS+LPI	FSSL	LPFL
2	93.63±15.30	92.14±15.77	92.98±14.28	93.58±13.14	97.20±4.00	92.23±13.35	98.72±2.24	**99.43±0.65**
3	89.62±16.15	88.02±18.60	88.16±15.03	88.46±16.45	90.55±10.44	85.72±14.93	94.39±10.98	**97.63±4.40**
4	86.59±18.79	84.30±20.15	88.41±9.12	86.38±16.72	90.95±9.16	86.64±9.11	92.12±8.18	**95.38±5.89**
5	80.67±17.52	77.36±19.90	89.76±8.97	81.25±17.21	86.74±13.04	85.21±9.04	91.85±7.83	**94.53±6.02**
6	83.19±15.33	75.97±16.74	87.49±9.49	76.35±13.55	82.11±9.00	88.01±6.98	92.08±9.02	**94.29±5.94**
7	74.12±12.35	70.47±1.35	93.05±3.87	73.24±13.43	79.29±13.22	89.27±6.90	91.60±6.09	92.87±6.13
8	72.35±16.09	68.51±16.43	86.76±11.07	67.99±18.58	74.64±15.57	88.07±7.17	90.56±7.32	**92.75±7.02**
9	72.57±14.06	65.36±17.59	91.50±5.41	67.39±14.93	73.95±13.38	92.04±5.03	92.33±4.76	**93.02±4.49**
10	63.22±15.28	57.55±16.09	86.94±9.30	59.44±16.92	65.67±11.48	89.02±7.63	88.60±9.18	89.26±7.04
avg	79.55	75.52	89.45	77.12	82.34	88.47	92.47	94.35

Table 2: Clustering accuracy on the Reuters21578 dataset

c	Baseline	LSI	LPI	LS	SLPI	LS+LPI	FSSL	LPFL
2	79.53±15.00	79.76±15.93	82.82±13.47	80.40±15.60	86.22±11.96	77.92±15.93	85.57±12.27	**86.30±11.51**
3	72.28±16.52	70.93±15.75	82.81±13.53	68.60±17.22	81.32±15.25	82.69±13.50	83.71±10.58	**86.36±9.83**
4	65.10±13.91	63.19±14.27	77.73±12.44	65.39±13.91	73.67±13.67	74.21±15.68	76.42±14.09	**79.44±10.33**
5	54.49±22.29	56.25±21.02	70.77±14.58	54.89±21.60	64.30±17.74	71.45±11.68	71.39±17.06	72.23±15.50
6	56.53±18.98	53.20±17.82	69.26±11.83	55.27±18.66	60.11±16.54	68.63±11.85	68.25±12.31	71.12±12.14
7	54.20±14.28	51.93±14.11	69.62±10.66	50.15±13.82	60.83±16.29	71.41±12.28	71.61±12.13	**72.44±12.27**
8	38.57±14.28	36.23±14.91	66.00±19.17	37.57±13.65	53.33±21.44	70.89±17.62	72.10±17.26	70.25±18.93
9	41.74±19.01	40.42±18.60	62.53±15.23	40.96±18.02	48.26±22.11	66.53±11.10	67.94±11.43	**69.71±11.98**
10	46.22±14.41	44.93±14.41	62.24±15.16	43.76±13.86	53.84±19.03	63.33±14.47	64.87±17.18	**67.00±15.61**
avg	56.52	55.20	71.53	55.22	64.65	71.90	73.57	74.98

preserving the locality of the data, LPI is able to discover the unbalanced clusters while K-means and LSI fail.

- Laplacian score usually performs worse than LPI. Since Laplacian score and LPI are based on the same criterion, this implies that feature combination is generally more effective than feature selection. SLPI is comparable to, or sometimes worse than LPI, which indicates that sparsity does not necessarily lead to performance improvement.

- In some cases, LS+LPI does achieve better results than standalone LPI. This highlights the potential performance gain of incorporating feature selection into LPI. But note that since Laplacian score and LPI are performed sequentially in LS+LPI, the features selected by Laplacian score are not necessarily optimal for LPI, resulting in limited improvement in performance.

- LPFL outperforms LPI, Laplacian score, SLPI and LS+LPI in most cases, showcasing the effectiveness of incorporating feature selection into LPI. Considered another way, this also indicates that Laplacian score and LPI can mutually enhance each other. Note that the average improvement of clustering accuracy over LS+LPI is from 88.47% to 94.35% on the TDT2 dataset, which is arguably better. It indicates that the features selected by LPFL should be inherently more useful than those selected by Laplacian score alone, because the feature selection and transformation are jointly optimized in LPFL.

- Finally, LPFL is better than FSSL, though FSSL performs very well in many cases. Despite of sharing similar spirit with LPFL, FSSL is optimized in an approximate way, which leads to performance sacrifice.

6 Conclusions and Future Work

In this paper, we propose an approach called *Locality Preserving Feature Learning*, which incorporates feature selection into LPI. We aim to find a subset of features, and a projection such that the *Locality Preserving Criterion* is minimized. The resulting optimization problem is relaxed into a constrained F-norm minimization problem and solved by applying a variation of Alternating Direction Method (ADM). Experiments on benchmark document datasets illustrate the efficacy of the proposed framework.

Although we have only studied document clustering in this paper, LPFL could be used for other kinds of document analysis as well. Moreover, as shown in Theorem 3, there could be a potential guidance to choose an appropriate or even optimal m for the proposed method. We will study these issues in the future work.

Acknowledgements

The work was supported in part by NSF IIS-09-05215, U.S. Air Force Office of Scientific Research MURI award FA9550-08-1-0265, and the U.S. Army Research Laboratory under Cooperative Agreement Number W911NF-09-2-0053 (NS-CTA). We thank the anonymous reviewers for their helpful comments.

References

[1] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

[2] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

[3] D. Cai and X. He. Orthogonal locality preserving indexing. In *SIGIR*, pages 3–10, 2005.

[4] D. Cai, X. He, and J. Han. Document clustering using locality preserving indexing. *IEEE Trans. Knowl. Data Eng.*, 17(12):1624–1637, 2005.

[5] D. Cai, X. He, and J. Han. Spectral regression: A unified approach for sparse subspace learning. In *ICDM*, pages 73–82, 2007.

[6] D. Cai, X. He, W. V. Zhang, and J. Han. Regularized locality preserving indexing via spectral regression. In *CIKM*, pages 741–750, 2007.

[7] F. R. K. Chung. *Spectral Graph Theory*. American Mathematical Society, 1997.

[8] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *JASIS*, 41(6):391–407, 1990.

[9] K. Fukunaga. *Introduction to statistical pattern recognition (2nd ed.)*. Academic Press Professional, Inc., 1990.

[10] R. Glowinski and P. L. Tallec. *Augmented Lagrangian and Operator-Splitting Methods in Nonlinear Mechanics*. SIAM Studies in Applied and Numerical Mathematics, 1989.

[11] Q. Gu, Z. Li, and J. Han. Joint feature selection and subspace learning. In *IJCAI*, pages 1294–1299, 2011.

[12] Q. Gu, Z. Li, and J. Han. Linear discriminant dimensionality reduction. In *ECML/PKDD (1)*, pages 549–564, 2011.

[13] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

[14] X. He, D. Cai, and P. Niyogi. Laplacian score for feature selection. In *NIPS*, 2005.

[15] X. He and P. Niyogi. Locality preserving projections. In *NIPS*, 2003.

[16] H. Lu, X. Long, and J. Lv. A fast algorithm for recovery of jointly sparse vectors based on the alternating direction methods. In *AISTATS*, 2011.

[17] M. Masaeli, G. Fung, and J. G. Dy. From transformation-based dimensionality reduction to feature selection. In *ICML*, pages 751–758, 2010.

[18] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620, 1975.

[19] W. Xu, X. Liu, and Y. Gong. Document clustering based on non-negative matrix factorization. In *SIGIR*, pages 267–273, 2003.

[20] S. Yan, D. Xu, B. Zhang, H. Zhang, Q. Yang, and S. Lin. Graph embedding and extensions: A general framework for dimensionality reduction. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(1):40–51, 2007.

[21] J. Yang and Y. Zhang. Alternating direction algorithms for l1-problems in compressive sensing. *SIAM Journal on Scientific Computing*, 2010.

[22] J. Ye. Least squares linear discriminant analysis. In *ICML*, pages 1087–1093, 2007.

[23] D. Zhang, X. Chen, and W. S. Lee. Text classification with kernels on the multinomial manifold. In *SIGIR*, pages 266–273, 2005.