Robust Unsupervised Feature Selection

Mingjie Qian and **Chengxiang Zhai**
Department of Computer Science
University of Illinois at Urbana-Champaign, IL, USA
{mqian2, czhai}@illinois.edu

Abstract

A new unsupervised feature selection method, i.e., Robust Unsupervised Feature Selection (RUFS), is proposed. Unlike traditional unsupervised feature selection methods, pseudo cluster labels are learned via local learning regularized robust nonnegative matrix factorization. During the label learning process, feature selection is performed simultaneously by robust joint $l_{2,1}$ norms minimization. Since RUFS utilizes $l_{2,1}$ norm minimization on processes of both label learning and feature learning, outliers and noise could be effectively handled and redundant or noisy features could be effectively reduced. Our method adopts the advantages of robust nonnegative matrix factorization, local learning, and robust feature learning. In order to make RUFS be scalable, we design a (projected) limited-memory BFGS based iterative algorithm to efficiently solve the optimization problem of RUFS in terms of both memory consumption and computation complexity. Experimental results on different benchmark real world datasets show the promising performance of RUFS over the state-of-the-arts.

1 Introduction

Data is often represented by high dimensional feature vectors in many areas, such as pattern recognition, text mining, computer vision [Wang *et al.*, 2009a][Wang *et al.*, 2009b], and bio-informatics. In practice, not all features are relevant and important to the learning task, many of them are often correlated, redundant, or even noisy sometimes [Duda *et al.*,][Huawei *et al.*, 2011], which may result in adverse effects such as over-fitting, low efficiency and poor performance. It is therefore important and necessary to reduce dimensionality. This can be usually achieved by transformation to a low dimensional space [Nie *et al.*, 2010b][Gu *et al.*, 2011] or feature selection. In this paper, we focus on feature selection, which aims to select discriminative and highly related features and eliminate unrelated, redundant, and noisy features with little or no supervision based on certain criteria.

During recent years, feature selection has attracted increasing attention, and many feature selection algorithms have been proposed, which can be grouped into three families: filter, wrapper, and embedded methods. Filter methods [Hall, 1999][Duda *et al.*,][He *et al.*, 2006][Zhao and Liu, 2007][Masaeli *et al.*, 2010][Huawei *et al.*, 2011][Yang *et al.*, 2011] select a subset of features by leveraging statistical properties of data, and are usually performed before applying classification algorithms. For wrapper methods [Kohavi and John, 1997][Guyon and Elisseeff, 2003][Rakotomamonjy, 2003], feature selection is wrapped in a learning algorithm and the classification performance on selected features is taken as the evaluation criterion. Embedded approaches [Vapnik, 1999][Zhu *et al.*, 2004][Hou *et al.*, 2011] perform feature selection when training the models. Wrapper and embedded methods couples feature selection with built-in classifiers tightly, which lead to less generality and extensive computation. We thus adopt the filter approach in this paper.

From the perspective of label availability, feature selection algorithms can also be classified into supervised feature selection and unsupervised feature selection. Supervised feature selection methods, such as [Duda *et al.*,][Nie *et al.*, 2010a][Zhao *et al.*, 2010][Nie *et al.*, 2008], are usually able to effectively select good features since labels of training data, which contain the essential discriminative information for classification, can be used. However, in unsupervised scenario, label information is unavailable directly, which makes the task of feature selection more challenging.

Several unsupervised feature selection algorithms are proposed recently. A commonly used criterion in unsupervised feature learning is to select features best preserving data similarity or manifold structure constructed from the whole feature space [He *et al.*, 2006][Zhao and Liu, 2007][Cai *et al.*, 2010], but they fail to incorporate discriminative information implied within data, though it has been shown to be important in data analysis [Fukunaga, 1990]. Earlier unsupervised feature selection algorithms evaluate the importance of each feature individually and select feature one by one [He *et al.*, 2006][Zhao and Liu, 2007], with a limitation that correlation among features is neglected pointed by [Zhao *et al.*, 2010][Cai *et al.*, 2010] which applied two-step approaches, i.e., spectral regression to unsupervised feature selection. State-of-the-arts unsupervised feature selection algorithms perform feature selection by simultaneously exploiting discriminative information and feature correlation. Unsupervised Discriminative Feature Selection (UDFS) [Yang *et al.*, 2011] aims to select the most discriminative features for data representation, where manifold structure is also considered. However, its orthogonal constraint on the feature selection projection matrix is unreasonable since feature weight vectors are not necessarily orthogonal with each other in nature. Nonnegative Discriminative Feature Selection (NDFS) [Li *et al.*, 2012] performs nonnegative spectral analysis and feature selection simultaneously. One factor that is ignored in both UDFS and NDFS is that data is usually not ideally clean, and outliers or noise often exist in it. UDFS and NDFS are not robust and are vulnerable to outliers or noise. Another deficiency of UDFS and NDFS is that their computation complexity is cubic to the number of features which severely limits their applicability on high dimensional data, e.g., text data and genetic data.

Since the most discriminative information for feature selection is usually encoded in labels, it is very important to predict a good cluster indicators as pseudo labels for unsupervised feature selection [Li *et al.*, 2012]. Another important factor which effects the performance of feature selection is the consideration of outliers and noise [Nie *et al.*, 2010a]. Real data is usually not ideally distributed, outliers and noise often appear in the data, thus it is important or even necessary to consider robustness for unsupervised feature selection.

In light of all these factors, we propose a new unsupervised feature selection algorithm, i.e., Robust Unsupervised Feature Selection (RUFS). We perform robust clustering and robust feature selection simultaneously to select the most important and discriminative features for unsupervised learning. Specifically, cluster indicators are generated by local learning regularized robust nonnegative matrix factorization, which is also a novel robust clustering method itself but we focus on unsupervised feature selection in this paper. Local learning [Gu and Zhou, 2009] is used in robust clustering which incorporates both discriminative information and the geometric structure and is good at clustering data on manifold. We impose an orthogonal constraint on the cluster indicator matrix to ensure that the learned cluster indicators are much closer to the true cluster labels. We then simultaneously perform robust feature selection using learned cluster indicators. RUFS exploits the discriminative information and feature correlation in a joint framework. Aiming at feature selection, joint $l_{2,1}$ norms minimization is utilized to learn a robust feature selection matrix which is sparse in rows. In order for the proposed RUFS be practical for real world applications, we present a (projected) limited-memory BFGS based iterative algorithm to solve the optimization problem of RUFS. Experiments on different real world datasets show that the RUFS outperforms the state-of-the-arts.

To summarize, the main contributions of this paper include:

- A Robust Unsupervised Feature Selection (RUFS) algorithm is proposed, where robust clustering and robust feature selection are simultaneously performed.
- A local learning regularized robust nonnegative matrix factorization with orthogonal constraint is proposed to learn the pseudo cluster indicator labels, which encodes both discriminative information and the geometric structure on manifold and eliminate the adverse effect of outliers and noise.
- A (projected) limited-memory BFGS based iterative algorithm is presented to efficiently solve RUFS in terms of both memory cost and computation complexity.

2 Notations and Preliminaries

Throughout this paper, matrices are written as boldface capital letters and vectors are denoted as boldface lowercase letters. For matrix $\mathbf{M} = (m_{ij})$, its i-th row, j-th column are denoted by $\mathbf{m}^i, \mathbf{m}_j$ respectively. $\|\mathbf{M}\|_F$ is the Frobenius norm of $\mathbf{M}$ and $\mathrm{Tr}\,[\mathbf{M}]$ is the trace of $\mathbf{M}$ if $\mathbf{M}$ is square. For any matrix $\mathbf{M} \in \mathcal{R}^{r\times t}$, its $l_{2,1}$-norm is defined as

$$\|\mathbf{M}\|_{2,1} = \sum_{i=1}^{r} \sqrt{\sum_{j=1}^{p} m_{ij}^2} = \sum_{i=1}^{r} \left\|\mathbf{m}^i\right\|_2. \tag{1}$$

Assume that we have n samples $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^n$. Let $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_n]^T$ denote the data matrix with each row being a data feature vector, in which $\mathbf{x}_i \in \mathcal{R}^d$ is the feature descriptor of the i-th sample. Suppose these n data samples are sampled from c classes and denote $\mathbf{Y} = [\mathbf{y}_1, \cdots, \mathbf{y}_n]^T \in \{0,1\}^{n\times c}$, where $\mathbf{y}_n \in \{0,1\}^{c\times 1}$ is the cluster indicator vector for sample $\mathbf{x}_i$. The scaled cluster indicator matrix [Yang *et al.*, 2011][Li *et al.*, 2012] $\mathbf{G}$ is defined as

$$\mathbf{G} = [\mathbf{g}_1, \cdots, \mathbf{g}_n]^T = \mathbf{Y}\left(\mathbf{Y}^T\mathbf{Y}\right)^{-\frac{1}{2}}, \tag{2}$$

where $\mathbf{g}_i$ is the scaled cluster indicator of $\mathbf{x}_i$. We thus have

$$\mathbf{G}^T\mathbf{G} = \left(\mathbf{Y}^T\mathbf{Y}\right)^{-\frac{1}{2}}\mathbf{Y}^T\mathbf{Y}\left(\mathbf{Y}^T\mathbf{Y}\right)^{-\frac{1}{2}} = \mathbf{I}_c, \tag{3}$$

where $\mathbf{I}_c \in \mathcal{R}^{c\times c}$ is an identity matrix.

2.1 Local learning regularization

According to [Bottou and Vapnik, 1992], searching a good predictor f in a global way might not be a good strategy because the function set $f(\mathbf{x})$ may not contain a good predictor for the entire input space. However, it is much easier to produce good predictions on some local regions of the input space and it is usually more effective to minimize prediction cost for each region. [Gu and Zhou, 2009] introduced a good way to construct the local predictors, and we will use it as our local learning regularization term.

Denoting $\mathcal{N}(\mathbf{x}_i)$ as the neighborhood of $\mathbf{x}_i$, the local learning regularization aims to minimize the sum of prediction errors between the local prediction from $\mathcal{N}(\mathbf{x}_i)$ and the cluster assignment of $\mathbf{x}_i$:

$$\begin{aligned} J &= \sum_{k=1}^{K}\sum_{i=1}^{n} \left\|f_i^k(\mathbf{x}_i) - g_{ik}\right\| \\ &= \sum_{k=1}^{K}\sum_{i=1}^{n} \left\|\mathbf{k}_i^T(\mathbf{K}_i + n_i\lambda\mathbf{I})^{-1}\mathbf{g}_i^k - g_{ik}\right\| \\ &= \sum_{k=1}^{K}\sum_{i=1}^{n} \left\|\alpha_i^T\mathbf{g}_i^k - g_{ik}\right\| \\ &= \mathrm{Tr}\left[\mathbf{G}^T\mathbf{L}\mathbf{G}\right] \end{aligned} \tag{4}$$

where $f_i^k(\mathbf{x}_i)$ is the locally predicted label for k-th cluster from $\mathcal{N}(\mathbf{x}_i)$, λ is a positive parameter, $\mathbf{K}_i$ is the kernel matrix defined on the neighborhood of $\mathbf{x}_i$, i.e., $\mathcal{N}(\mathbf{x}_i)$, with size of n_i, $\mathbf{k}_i$ is the kernel vector defined between $\mathbf{x}_i$ and $\mathcal{N}(\mathbf{x}_i)$, $\mathbf{g}_i^k$ is the cluster assignments of $\mathcal{N}(\mathbf{x}_i)$, $\mathbf{I} \in \mathcal{R}^{n\times n}$ is an identity matrix, and $\mathbf{A} \in \mathcal{R}^{n\times n}$ is defined by

$$\mathbf{A}_{ij} = \left\{ \begin{array}{ll} \alpha_{ij}, & \text{if } \mathbf{x}_j \in \mathcal{N}(\mathbf{x}_i) \\ 0, & \text{otherwise} \end{array} \right. . \qquad (5)$$

3 The Objective Function

In this section, we present the objective function of the proposed Robust Unsupervised Feature Selection (RUFS) algorithm. To select discriminative features for unsupervised learning, learning accurate pseudo cluster labels are very important. NDFS [Li *et al.*, 2012] uses spectral clustering to predict the labels. In this work, however, we propose to utilize local learning regularized robust nonnegative matrix factorization with orthogonal constraint to learn the pseudo cluster labels. The reason is three folds. First, since spectral clustering relies on similarity matrix computed from the original feature space. Thus unrelated or noisy features will have an adverse effect on clustering and henceforth hurt feature selection performance. Although nonnegative matrix factorization [Lee *et al.*, 1999] decomposes the data matrix in the original feature space, the adverse effect will be mainly accumulated in the learned cluster centers and won't hurt the indicators severely. Second, a robust clustering algorithm that considers outliers and noise usually improves the clustering performance [Kong *et al.*, 2011]. Third, researchers [Bottou and Vapnik, 1992][Wu and Scholkopf, 2007][Gu and Zhou, 2009] have shown that local learning is more effective than learning a good predictor in a global way since the function set may not contain a good predictor for the entire input space. Thus we use the local learning regularization to encode the discriminative information and the geometric structure via local predictors, which results in good clustering performance particularly on data embedded on manifold.

Our proposed robust clustering via local learning differs from [Kong *et al.*, 2011] in that local learning is involved during the clustering procedure and standardized orthogonal constraint is imposed on the indicator matrix so that arbitrary scaling and trivial solutions could be avoided and more ideal pseudo cluster labels could be learned.

Given the proposed robust clustering with local learning, RUFS aims to solve the following optimization problem:

$$\begin{aligned} \min_{\mathbf{F},\mathbf{G},\mathbf{W}} \quad & \|\mathbf{X}-\mathbf{G}\mathbf{F}\|_{2,1} + \nu \mathrm{Tr}\left[\mathbf{G}^T\mathbf{L}\mathbf{G}\right] + \\ & \alpha\|\mathbf{X}\mathbf{W}-\mathbf{G}\|_{2,1} + \beta\|\mathbf{W}\|_{2,1} \\ \text{s.t.} \quad & \mathbf{G}\in\mathcal{R}_+^{n\times c}, \mathbf{G}=\mathbf{Y}\left(\mathbf{Y}^T\mathbf{Y}\right)^{-\frac{1}{2}}, \mathbf{F}\in\mathcal{R}_+^{c\times d}, \end{aligned} \qquad (6)$$

where $\nu, \alpha, \beta \in \mathcal{R}_+$ are parameters. Robust feature selection is performed through jointly minimizing the last two terms (joint $l_{2,1}$ norms minimization), which is able to handle outliers and noise in data. The $l_{2,1}$ norm imposed on the feature selection matrix $\mathbf{W}$ guarantees the property of sparseness in rows. More specifically, $\mathbf{w}^j$ shrinks to zero if the j-th feature is less correlated to the pseudo labels $\mathbf{G}$. We can thus filter out the features corresponding to zero rows of $\mathbf{W}$ when performing feature selection.

Since $\mathbf{Y}$ by definition is a 0 or 1 matrix, the optimization of Eq. (6) is an NP-hard problem [Shi and Malik, 2000]. A commonly used strategy is to relax it to continuous values while keeping the key property, we thus constraint $\mathbf{G}$ to be orthonormal by columns, and the original optimization problem is relaxed to

$$\begin{aligned} \min_{\mathbf{F},\mathbf{G},\mathbf{W}} \quad & \|\mathbf{X}-\mathbf{G}\mathbf{F}\|_{2,1} + \nu \mathrm{Tr}\left[\mathbf{G}^T\mathbf{L}\mathbf{G}\right] + \\ & \alpha\|\mathbf{X}\mathbf{W}-\mathbf{G}\|_{2,1} + \beta\|\mathbf{W}\|_{2,1} \\ \text{s.t.} \quad & \mathbf{G}\in\mathcal{R}_+^{n\times c}, \mathbf{G}^T\mathbf{G}=\mathbf{I}_c, \\ & \mathbf{F}\in\mathcal{R}_+^{c\times d}, \mathbf{W}\in\mathcal{R}^{d\times c}, \end{aligned} \qquad (7)$$

where the first two terms learn the pseudo cluster labels using robust orthogonal nonnegative matrix factorization via local learning regularization while the last two terms simultaneously learn the feature selection matrix by joint $l_{2,1}$ norms minimization.

By solving optimization problem (7), we learn three components of the robust unsupervised feature selection model, i.e., the pseudo cluster labels $\mathbf{G}$ which is very close to the ideal scaled label indicators, the cluster centers $\mathbf{F}$ in the original whole feature space, and the feature selection matrix (or projection matrix for regression) $\mathbf{W}$ which is sparse in rows.

4 Optimization Algorithm

In the era of big data, high dimensional data is prevalent and the number of features is usually very high (otherwise, we may not need feature selection), for example, text data, genetic data, or image data with high resolution. In such cases, both UDFS and NDFS will be prohibitively slow since they share the computation complexity of $O\left(d^3\right)$ and memory complexity of $O\left(d^2\right)$. For practical use of unsupervised feature selection, we require algorithms to be able to handle large number of features and large number of data samples which are not only computionally efficient but also save memory.

Limited-memory quasi-Newton methods [Nocedal and Wright, 1999] are among the best candidates for solving large scale optimization problems when Hessian matrices cannot be computed at a reasonable cost or are not sparse. These methods maintain simple and compact approximations of Hessian matrices using only a few vectors that represent the approximations implicitly. Despite these modest storage requirements, they often yield an acceptable (albeit linear) rate of convergence. In this section, we present an iterative algorithm to efficiently solve Eq. (7) using L-BFGS [Liu and Nocedal, 1989] and projected L-BFGS [Benson and More, 2001] methods.

To solve RUFS, we first rewrite the optimization problem as follows

$$\begin{aligned} \min_{\mathbf{F},\mathbf{G},\mathbf{W}} \quad & \|\mathbf{X}-\mathbf{G}\mathbf{F}\|_{2,1} + \nu \mathrm{Tr}\left[\mathbf{G}^T\mathbf{L}\mathbf{G}\right] + \beta\|\mathbf{W}\|_{2,1} + \\ & \alpha\|\mathbf{X}\mathbf{W}-\mathbf{G}\|_{2,1} + \frac{\zeta}{4}\left\|\mathbf{G}^T\mathbf{G}-\mathbf{I}_c\right\|_F^2 \\ \text{s.t.} \quad & \mathbf{G}\in\mathcal{R}_+^{n\times c}, \mathbf{F}\in\mathcal{R}_+^{c\times d}, \mathbf{W}\in\mathcal{R}^{d\times c} \end{aligned} \qquad (8)$$

where ζ is a parameter to control the orthogonality condition. In practice, ζ should be large enough to insure the orthogonality satisfied. We first define the objective function

$$\mathscr{L}(\mathbf{G},\mathbf{F},\mathbf{W}) = \|\mathbf{X}-\mathbf{G}\mathbf{F}\|_{2,1} + \nu \mathrm{Tr}\left[\mathbf{G}^T\mathbf{L}\mathbf{G}\right] + \\ \alpha\|\mathbf{X}\mathbf{W}-\mathbf{G}\|_{2,1} + \beta\|\mathbf{W}\|_{2,1} + \frac{\zeta}{4}\left\|\mathbf{G}^T\mathbf{G}-\mathbf{I}_c\right\|_F^2, \quad (9)$$

denoting

$$\begin{aligned} \mathbf{r}_1 &= \left[\left\|\mathbf{x}^1-\mathbf{g}^1\mathbf{F}\right\|_2,\ldots,\left\|\mathbf{x}^n-\mathbf{g}^n\mathbf{F}\right\|_2\right]^T, \\ \mathbf{r}_2 &= \left[\left\|\mathbf{x}^1\mathbf{W}-\mathbf{g}^1\right\|_2,\ldots,\left\|\mathbf{x}^n\mathbf{W}-\mathbf{g}^n\right\|_2\right]^T, \\ \mathbf{r}_3 &= \left[\left\|\mathbf{w}^1\right\|_2,\ldots,\left\|\mathbf{w}^d\right\|_2\right]^T, \end{aligned}$$

the partial derivatives of $\mathscr{L}(\mathbf{G},\mathbf{F},\mathbf{W})$ w.r.t. $\mathbf{G},\mathbf{F}$, and $\mathbf{W}$ can be obtained

$$\begin{aligned} \nabla_{\mathbf{G}}\mathscr{L} &= (\mathbf{G}\mathbf{F}-\mathbf{X})\,\mathbf{F}^T \oslash [\mathbf{r}_1 \otimes \mathbb{1}_{1\times c}] + 2\nu\mathbf{L}\mathbf{G} + \\ &\quad \alpha(\mathbf{G}-\mathbf{X}\mathbf{W}) \oslash [\mathbf{r}_2 \otimes \mathbb{1}_{1\times c}] + \\ &\quad \zeta\mathbf{G}\left(\mathbf{G}^T\mathbf{G}-\mathbf{I}_c\right), & (10) \\ \nabla_{\mathbf{F}}\mathscr{L} &= \mathbf{G}^T\left[(\mathbf{G}\mathbf{F}-\mathbf{X}) \oslash [\mathbf{r}_1 \otimes \mathbb{1}_{1\times d}]\right], & (11) \\ \nabla_{\mathbf{W}}\mathscr{L} &= \alpha\mathbf{X}^T\left[(\mathbf{X}\mathbf{W}-\mathbf{G}) \oslash [\mathbf{r}_2 \otimes \mathbb{1}_{1\times c}]\right] + \\ &\quad \beta\mathbf{W} \oslash [\mathbf{r}_3 \otimes \mathbb{1}_{1\times c}], & (12) \end{aligned}$$

where $\otimes$ is the Kronecker product, $\oslash$ is the element-wise division, and $\mathbb{1}$ is an all 1 matrix. Solutions of problem (8) satisfy the Kuhn-Tucker conditions

$$\begin{cases} \frac{\partial\mathscr{L}}{\partial G_{ik}} = 0 \text{ if } G_{ik} > 0;\ \frac{\partial\mathscr{L}}{\partial G_{ik}} \geq 0 \text{ if } G_{ik} = 0 \\ \frac{\partial\mathscr{L}}{\partial F_{kj}} = 0 \text{ if } F_{kj} > 0;\ \frac{\partial\mathscr{L}}{\partial F_{kj}} \geq 0 \text{ if } F_{kj} = 0 \\ \frac{\partial\mathscr{L}}{\partial W_{jk}} = 0 \end{cases} . \quad (13)$$

The projection operator

$$[T_\Omega\mathbf{M}]_{ij} = \begin{cases} M_{ij} & \text{if} \quad X_{ij} > 0 \\ \min\{M_{ij}, 0\} & \text{if} \quad X_{ij} = 0 \end{cases} \quad (14)$$

can be helpful because $(\mathbf{G}^*,\mathbf{F}^*,\mathbf{W}^*)$ is a solution of problem (8) if and only if

$$\left(T_{\mathcal{R}_+^{n\times c}}\nabla\mathbf{G}^*, T_{\mathcal{R}_+^{c\times d}}\nabla\mathbf{F}^*, \nabla\mathbf{W}^*\right) = \mathbf{0}. \quad (15)$$

Given a tolerance τ, an approximate solution to problem (8) is any matrix triplet $(\mathbf{G},\mathbf{F},\mathbf{W})$ such that

$$\left\|\left(T_{\mathcal{R}_+^{n\times c}}\nabla\mathbf{G}, T_{\mathcal{R}_+^{c\times d}}\nabla\mathbf{F}, \nabla\mathbf{W}\right)\right\| \leq \tau. \quad (16)$$

In next subsection, we will present a limited-memory BFGS based alternating iterative algorithm to efficiently solve problem (8).

4.1 Limited-memory BFGS

Recall that each step of the BFGS method has the form

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k\mathbf{H}_k\nabla f_k, \quad (17)$$

where α_k is the step length and $\mathbf{H}_k$ is the inverse Hessian approximation. Since $\mathbf{H}_k$ will generally be dense, the cost of storing and manipulating it is prohibitive when the number of variables is large. To circumvent this problem, limited-memory BFGS computes a modified version of $\mathbf{H}_k$ implicitly by storing a certain number (say, m) of most recent correction pairs using a two-loop recursive procedure to compute the product $\mathbf{H}_k\nabla f$ efficiently [Nocedal and Wright, 1999]. The limited-memory BFGS algorithm can thus be stated formally shown in Algorithm 1. When nonnegative constraints are imposed, a projected version of limited-memory BFGS algorithm is required. There are various projected limited-memory BFGS algorithms, here we choose BMLVM algorithm [Benson and More, 2001] for its faster speed than L-BFGS-B [Byrd *et al.*, 1995] and ease of use.

Algorithm 1 L-BFGS

```
Input: Staring point $\mathbf{x}_0$ and an integer $m > 0$
$k \leftarrow 0$
repeat
    Compute $\mathbf{p}_k \leftarrow -\mathbf{H}_k\nabla f_k$ using a two-loop recursion
    Compute $\mathbf{x}_{k+1} \leftarrow \mathbf{x}_k + \alpha_k\mathbf{p}_k$, where $\alpha_k$ is chosen to
        satisfy the Wolfe conditions
    if $k > m$ then
        Discard the vector pair $\{\mathbf{s}_{k-m}, \mathbf{y}_{k-m}\}$
    end if
    Save $\mathbf{s}_k \leftarrow \mathbf{x}_{k+1} - \mathbf{x}_k, \mathbf{y}_k \leftarrow \nabla f_{k+1} - \nabla f_k$
    $k \leftarrow k + 1$
until $\|\nabla f_k\| \leq \tau$
Output: $\mathbf{x}_k$
```

4.2 RUFS Algorithm

We adopt an alternating optimization (AO) strategy to solve RUFS and list it in Algorithm 3. Following the convergence analysis for a general AO approach, the convergence of Algorithm 3 can be shown to be locally and q-linearly convergent [Bezdek and Hathaway, 2003].

4.3 Complexity Analysis

The two-loop recursion scheme requires O(4tmdc) for computing $\mathbf{W}$ and $\mathbf{F}$ and O(4tmnc) for computing $\mathbf{G}$, so we have O(4tm(2d + n)c) scaler multiplications where t is the total number of inner iterations of Algorithm 3. According to Eq. (10) $\sim$ Eq. (12), computing partial gradients w.r.t. $\mathbf{W}$, $\mathbf{F}$, and $\mathbf{G}$ are $t_G * O(3ndc + cn^2)$, $t_F * O(2ndc)$, and $t_W * O(2ndc)$ respectively, where t_A is the total number of inner iterations of Algorithm 3 computing matrix $\mathbf{A}$. Evaluation of objective function values requires about $\#lineSearchIter * O(2ncd + 2cn^2)$. Note that the $O(cn^2)$ part is inevitable due to the local learning regularization term (UDFS and NDFS also share the $O(cn^2)$ part due to local discriminative analysis). The computation of projection operation can be ignored compared to the computation of gradients and objective function values because only boolean operations are performed. The memory complexity of RUFS is $O(n^2)+O(c^2)+O(nc)+O(dc)+O(nd) \sim O(n^2)+O(nd)$, which is dominated by $O(n^2)$ when data matrix is sparse (e.g., for text data). Note that both UDFS and NDFS require

$O(d^2) + O(n^2)$ memory cost and $O(d^3) + O(cn^2)$ computation complexity, which will be prohibitively slow when d the original feature size is very large.

```
Algorithm 2 BMLVM
  Input: Staring point $\mathbf{x}_0$ and an integer $m > 0$
  $k \leftarrow 0$
  repeat
      Compute $\mathbf{p}_k \leftarrow -\mathbf{H}_k \nabla f_k$ using a two-loop recursion
      if $\langle T_\Omega(\mathbf{H}_k \nabla f_k), \nabla f_k \rangle > 0$ then
          $\mathbf{p}_k \leftarrow -T_\Omega(\mathbf{H}_k \nabla f_k)$
      else
          $\mathbf{p}_k \leftarrow -T_\Omega \nabla f_k$
      end if
      Compute $\mathbf{x}_{k+1} \leftarrow [\mathbf{x}_k + \alpha_k \mathbf{p}_k]_+$, where $\alpha_k$ is chosen
          to satisfy the Wolfe conditions
      if $k > m$ then
          Discard the vector pair $\{\mathbf{s}_{k-m}, \mathbf{y}_{k-m}\}$
      end if
      Save $\mathbf{s}_k \leftarrow \mathbf{x}_{k+1} - \mathbf{x}_k, \mathbf{y}_k \leftarrow T_\Omega \nabla f_{k+1} - T_\Omega \nabla f_k$
      $k \leftarrow k + 1$
  until $\|T_\Omega \nabla f_k\| \le \tau$
  Output: $\mathbf{x}_k$
```

```
Algorithm 3 RUFS
Input: $\mathbf{X} \in \mathcal{R}^{n \times d}, \nu, \alpha, \beta, c$, and $p$
  Construct $\mathbf{L}$ from Eq. (4)
  Initialize $\mathbf{G}_0$ (e.g., by K-means)
  Initialize $\mathbf{F_0} \leftarrow \left[\left(\mathbf{G^T G}\right)^{-1} \mathbf{G^T X}\right]_+$
  Initialize $\mathbf{W}_0$
  $k \leftarrow 0$
  repeat
      Fixing $\mathbf{G}_k$, compute $\mathbf{W}_{k+1}$ from
          Algorithm 1 given $\mathbf{G}_k, \mathbf{W}_k, \alpha, \beta$
      Fixing $\mathbf{F}_k$ and $\mathbf{W}_{k+1}$, compute $\mathbf{G}_{k+1}$ from
          Algorithm 2 given $\mathbf{G}_k, \mathbf{F}_k, \mathbf{W}_{k+1}\mathbf{L}, \nu$, and $\alpha$
      Fixing $\mathbf{G}_{k+1}$, compute $\mathbf{F}_{k+1}$ from
          Algorithm 2 given $\mathbf{G}_{k+1}, \mathbf{F}_k$
      $k \leftarrow k + 1$
  until $\left\| \left( T_{\mathcal{R}_+^{n \times c}} \nabla \mathbf{G}_k, T_{\mathcal{R}_+^{c \times d}} \nabla \mathbf{F}_k, \nabla \mathbf{W}_k \right) \right\| \le \tau$
Output: Sort all $d$ features according to $\|\mathbf{w}_k^j\|_2$ in descending
order and select the top $p$ ranked features.
```

5 Experiments

In this section, we conduct experiments to evaluate RUFS. Following previous unsupervised feature selection work [Cai *et al.*, 2010][Yang *et al.*, 2011][Li *et al.*, 2012], we only evaluate the performance of RUFS for feature selection on clustering due to space limit.

Table 1: Dataset Description.

Dataset	# of Samples	# of Features	# of Classes
ORL	400	1024	40
COIL20	1440	1024	20
BinaryAlphadigits	1404	320	36
UMIST	575	644	20
Isolet	1560	617	26
WebKB4	4199	1000	4

5.1 Datasets

The evaluation is performed on 6 benchmark real world datasets including ORL (AT&T)[1], COIL20[2], Binary Alphadigits[3], UMIST[4], Isolet1[5], and WebKB4 [Gu and Zhou, 2009]. Detailed information is summarized in Table 1.

5.2 Compared Methods

We compare RUFS with the following unsupervised feature selection algorithms.

1. **Baseline**: All original features are adopted.
2. **LS**: Laplacian Score [He *et al.*, 2006] which selects features that best preserve the local manifold structure.
3. **MCFS**: Mutli-Cluster Feature Selection [Cai *et al.*, 2010] where features are selected using spectral regression with l_1-norm regularization.
4. **UDFS**: Unsupervised Discriminative Feature Selection [Yang *et al.*, 2011] which exploits local discriminative information and feature correlations simultaneously and considers the manifold structure as well.
5. **NDFS**: Nonnegative Discriminative Feature Selection [Li *et al.*, 2012] where features are selected by a joint framework of nonnegative spectral analysis and $l_{2,1}$-norm regularized regression.

5.3 Experiment Setup

Following previous work, two widely used evaluation metrics, i.e., Accuracy (ACC) and Normalized Mutual Information (NMI) are used in this paper.

There are some parameters to be set. Following previous work, for LS, MCFS, UDFS, NDGS, and RUFS, we fix $k = 5$ for all the datasets to specify the neighborhood size. To fairly compare different unsupervised feature selection methods, we tune the parameters for all methods by a "grid-search" strategy from $\{10^{-6}, 10^{-4}, \ldots, 10^4, 10^6\}$. The number of selected features are set as $\{50, 100, 150, \ldots 300\}$ for all datasets. Best clustering results from the optimal parameters are reported for all the algorithms. In the evaluation, we use K-means to cluster samples based on the selected features. Since K-means depends on initialization, following previous work, we repeat clustering 20 times with random initialization for each setup. The average results with standard deviation are reported.

[1] http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html

[2] http://www.cad.zju.edu.cn/home/dengcai/Data/MLData.html

[3] http://www.cs.nyu.edu/∼roweis/data.html

[4] http://www.sheffield.ac.uk/eee/research/iel/research/face

[5] http://archive.ics.uci.edu/ml/datasets/ISOLET

Table 2: Clustering Results (ACC% $\pm$ std) of Different Feature Selection Algorithms.

Dataset	All Features	Laplacian Score	MCFS	UDFS	NDFS	RUFS
ORL	51.1 ± 3.2	47.2 ± 2.5	49.7 ± 3.7	51.3 ± 3.0	52.3 ± 3.2	$\mathbf{53.4 \pm 3.8}$
COIL20	60.4 ± 4.5	56.4 ± 4.6	60.9 ± 4.7	59.8 ± 4.4	59.7 ± 3.3	$\mathbf{62.0 \pm 3.2}$
BinaryAlphadigits	41.0 ± 2.1	42.3 ± 1.8	41.8 ± 2.3	42.4 ± 1.8	42.4 ± 1.7	$\mathbf{42.7 \pm 1.7}$
UMIST	41.7 ± 2.5	44.1 ± 2.7	45.4 ± 2.6	45.3 ± 2.4	48.2 ± 3.6	$\mathbf{49.1 \pm 3.2}$
Isolet	59.7 ± 3.6	56.2 ± 3.7	56.9 ± 4.7	56.2 ± 3.8	63.0 ± 4.6	$\mathbf{64.5 \pm 3.2}$
WebKB4	69.2 ± 8.6	49.1 ± 7.9	59.5 ± 9.6	60.1 ± 5.8	69.2 ± 6.7	$\mathbf{74.2 \pm 2.5}$

Table 3: Clustering Results (NMI% $\pm$ std) of Different Feature Selection Algorithms.

Dataset	All Features	Laplacian Score	MCFS	UDFS	NDFS	RUFS
ORL	74.0 ± 1.9	71.5 ± 1.2	73.7 ± 1.8	73.4 ± 1.6	74.9 ± 1.9	$\mathbf{75.1 \pm 1.8}$
COIL20	76.3 ± 1.8	71.8 ± 2.0	74.9 ± 2.2	74.7 ± 1.6	76.0 ± 1.6	$\mathbf{77.0 \pm 2.2}$
BinaryAlphadigits	57.6 ± 1.3	58.5 ± 0.9	58.3 ± 1.2	58.8 ± 0.9	58.6 ± 0.8	$\mathbf{59.4 \pm 1.0}$
UMIST	63.9 ± 1.8	65.9 ± 1.4	66.6 ± 1.7	65.2 ± 1.6	66.5 ± 2.2	$\mathbf{68.8 \pm 2.4}$
Isolet	75.9 ± 1.6	73.1 ± 1.5	73.1 ± 1.4	72.8 ± 1.8	78.6 ± 1.6	$\mathbf{78.9 \pm 1.1}$
WebKB4	46.7 ± 3.1	29.2 ± 11.5	37.4 ± 15.3	34.5 ± 5.2	45.3 ± 4.9	$\mathbf{49.5 \pm 2.9}$

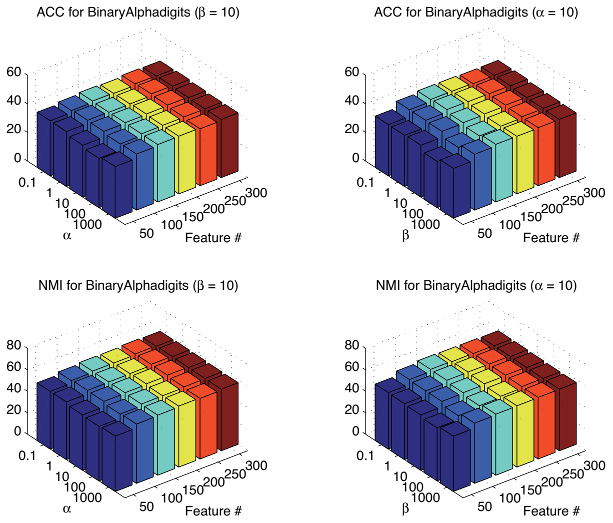

Figure 1: ACC and NMI of RUFS with different α, β and feature numbers while keeping $\nu = 10$.

Table 4: Average Running Time (seconds).

Dataset	UDFS	NDFS	RUFS
COIL20	42.4	50.4	**32.2**
WebKB4	112.9	281.3	**86.1**

5.4 Results and Discussion

We list the experimental results of different methods in Table 2 and Table 3. We observe from the clustering results that feature selection is important and effective. Not only can number of features be significantly reduced which makes posterior processing more efficient, but clustering performance can also be greatly improved. A new observation is that robust analysis is important for unsupervised learning. Consideration of outliers and noise usually improves the performance of feature selection, which has also been observed in supervised scenario. At last, RUFS achieves the best performance. This can be mainly explained by the following reasons. First, joint learning is performed between robust label learning and robust feature selection. Second, local learning is exploited which results in more accurate pseudo labels. Third, outliers and noise are considered during processes of both label learning and feature learning, so that more accurate and discriminative pseudo labels can be obtained.

We also study the sensitiveness of parameters. Due to space limit, we only report the results on BinaryAlphadigits dataset with $\nu = 10$ (sensitiveness under other values of ν is similar) on Figure 1. The experimental results show that our method is not very sensitive to α and β with wide ranges. However, the performance is relatively sensitive to the number of selected features, which is still an open problem.

We finally compare the running time of UDFS, NDFS, and RUFS in Table 4 on COIL20 and WebKB4 datasets (other datasets have either a smaller sample size or a smaller feature size). The calculations are performed using an Intel(R) Core(TM) i7 CPU M620 @ 2.67GHz with 4.00GB memory and 64-bit Windows 7 operating system. The empirical results in Table 4 are consistent with the theoretical analysis.

6 Conclusion

We propose a new robust unsupervised feature selection approach called RUFS, which jointly performs robust label learning via local learning regularized robust orthogonal nonnegative matrix factorization and robust feature learning via joint $l_{2,1}$-norms minimization. To make RUFS be applicable for large scale feature selection tasks, we present a (projected) limited-memory based iterative algorithm to solve it. Experimental results on different real world datasets validate the effectiveness of the proposed method.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant Number CNS-1027965.

References

[Benson and More, 2001] S.J. Benson and J.J. More. A limited memory variable metric method in subspaces and bound constrained optimization problems. *mathematical, Information and Computer Science Division, Argonne National Laboratory, ANL/MCS-P*, 901, 2001.

[Bezdek and Hathaway, 2003] J.C. Bezdek and R.J. Hathaway. Convergence of alternating optimization. *Neural, Parallel and Scientific Computations*, 11(4):351–368, 2003.

[Bottou and Vapnik, 1992] L. Bottou and V. Vapnik. Local learning algorithms. *Neural computation*, 4(6):888–900, 1992.

[Byrd *et al.*, 1995] R.H. Byrd, P. Lu, J. Nocedal, and C. Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16(5):1190–1208, 1995.

[Cai *et al.*, 2010] D. Cai, C. Zhang, and X. He. Unsupervised feature selection for multi-cluster data. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 333–342. ACM, 2010.

[Duda *et al.*,] RO Duda, PE Hart, and DG Stork. Pattern recognition. 2001.

[Fukunaga, 1990] Keinosuke Fukunaga. *Introduction to statistical pattern recognition*. Academic Pr, 1990.

[Gu and Zhou, 2009] Q. Gu and J. Zhou. Local learning regularized nonnegative matrix factorization. In *Twenty-First International Joint Conference on Artificial Intelligence*, 2009.

[Gu *et al.*, 2011] Quanquan Gu, Zhenhui Li, and Jiawei Han. Joint feature selection and subspace learning. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Volume Two*, pages 1294–1299. AAAI Press, 2011.

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

[Hall, 1999] Mark A Hall. *Correlation-based feature selection for machine learning*. PhD thesis, The University of Waikato, 1999.

[He *et al.*, 2006] X. He, D. Cai, and P. Niyogi. Laplacian score for feature selection. *Advances in Neural Information Processing Systems*, 18:507, 2006.

[Hou *et al.*, 2011] Chenping Hou, Feiping Nie, Dongyun Yi, and Yi Wu. Feature selection via joint embedding learning and sparse regression. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Volume Two*, pages 1324–1329. AAAI Press, 2011.

[Huawei *et al.*, 2011] L. Huawei, W. Xindong, and Z. Shichao. Feature selection using hierarchical feature clustering. 2011.

[Kohavi and John, 1997] Ron Kohavi and George H John. Wrappers for feature subset selection. *Artificial intelligence*, 97(1):273–324, 1997.

[Kong *et al.*, 2011] D. Kong, C. Ding, and H. Huang. Robust nonnegative matrix factorization using l21-norm. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 673–682. ACM, 2011.

[Lee *et al.*, 1999] D.D. Lee, H.S. Seung, et al. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[Li *et al.*, 2012] Z. Li, Y. Yang, J. Liu, X. Zhou, and H. Lu. Unsupervised feature selection using nonnegative spectral analysis. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[Liu and Nocedal, 1989] D.C. Liu and J. Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1):503–528, 1989.

[Masaeli *et al.*, 2010] M. Masaeli, G. Fung, and J.G. Dy. From transformation-based dimensionality reduction to feature selection. In *Int. Conf. on Machine Learning, Citeseer*, 2010.

[Nie *et al.*, 2008] F. Nie, S. Xiang, Y. Jia, C. Zhang, and S. Yan. Trace ratio criterion for feature selection. In *Proceedings of the 23rd national conference on Artificial intelligence*, volume 2, pages 671–676, 2008.

[Nie *et al.*, 2010a] F. Nie, H. Huang, X. Cai, and C. Ding. Efficient and robust feature selection via joint l2, 1-norms minimization. *Advances in Neural Information Processing Systems*, 23:1813–1821, 2010.

[Nie *et al.*, 2010b] Feiping Nie, Dong Xu, IW-H Tsang, and Changshui Zhang. Flexible manifold embedding: a framework for semi-supervised and unsupervised dimension reduction. *Image Processing, IEEE Transactions on*, 19(7):1921–1932, 2010.

[Nocedal and Wright, 1999] J. Nocedal and S.J. Wright. *Numerical optimization*. Springer verlag, 1999.

[Rakotomamonjy, 2003] A. Rakotomamonjy. Variable selection using svm based criteria. *The Journal of Machine Learning Research*, 3:1357–1370, 2003.

[Shi and Malik, 2000] J. Shi and J. Malik. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):888–905, 2000.

[Vapnik, 1999] V. Vapnik. *The nature of statistical learning theory*. springer, 1999.

[Wang *et al.*, 2009a] M. Wang, X.S. Hua, R. Hong, J. Tang, G.J. Qi, and Y. Song. Unified video annotation via multigraph learning. *Circuits and Systems for Video Technology, IEEE Transactions on*, 19(5):733–746, 2009.

[Wang *et al.*, 2009b] M. Wang, X.S. Hua, J. Tang, and R. Hong. Beyond distance measurement: constructing neighborhood similarity for video annotation. *Multimedia, IEEE Transactions on*, 11(3):465–476, 2009.

[Wu and Scholkopf, 2007] M. Wu and B. Scholkopf. A local learning approach for clustering. *Advances in neural information processing systems*, 19:1529, 2007.

[Yang *et al.*, 2011] Y. Yang, H.T. Shen, Z. Ma, Z. Huang, and X. Zhou. l 2, 1-norm regularized discriminative feature selection for unsupervised learning. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Volume Two*, pages 1589–1594. AAAI Press, 2011.

[Zhao and Liu, 2007] Z. Zhao and H. Liu. Spectral feature selection for supervised and unsupervised learning. In *Proceedings of the 24th international conference on Machine learning*, pages 1151–1157. ACM, 2007.

[Zhao *et al.*, 2010] Z. Zhao, L. Wang, and H. Liu. Efficient spectral feature selection with minimum redundancy. In *Proceedings of the Twenty-4th AAAI Conference on Artificial Intelligence (AAAI)*, 2010.

[Zhu *et al.*, 2004] J. Zhu, S. Rosset, T. Hastie, and R. Tibshirani. 1-norm support vector machines. *Advances in neural information processing systems*, 16(1):49–56, 2004.