Embedded Unsupervised Feature Selection

Suhang Wang, Jiliang Tang, Huan Liu
School of Computing, Informatics, and Decision Systems Engineering
Arizona State University, USA
{ suhang.wang, jiliang.tang, huan.liu } @ asu.edu

Abstract
Sparse learning has been proven to be a powerful technique in supervised feature selection, which allows to embed feature selection into the classification (or regression) problem. In recent years, increasing attention has been on applying sparse learning in unsupervised feature selection. Due to the lack of label information, the vast majority of these algorithms usually generate cluster labels via clustering algorithms and then formulate unsupervised feature selection as sparse learning based supervised feature selection with these generated cluster labels. In this paper, we propose a novel unsupervised feature selection algorithm EUFS, which directly embeds feature selection into a clustering algorithm via sparse learning without the transformation. The Alternating Direction Method of Multipliers is used to address the optimization problem of EUFS. Experimental results on various benchmark datasets demonstrate the effectiveness of the proposed framework EUFS.

Introduction
In many real-world applications such as data mining and machine learning, one is often faced with high-dimensional data (Jain and Zongker 1997; Guyon and Elisseeff 2003). Data with high dimensionality not only significantly increases the time and memory requirements of the algorithms, but also degrades many algorithms’ performance due to the curse of dimensionality and the existence of irrelevant, redundant and noisy dimensions (Liu and Motoda 2007). Feature selection, which reduces the dimensionality by selecting a subset of most relevant features, has been proven to be an effective and efficient way to handle high-dimensional data (John et al. 1994; Liu and Motoda 2007).

In terms of the label availability, feature selection methods can be broadly classified into supervised methods and unsupervised methods. The availability of the class label allows supervised feature selection algorithms (Duda et al. 2001; Nie et al. 2008; Zhao et al. 2010; Tang et al. 2014) to effectively select discriminative features to distinguish samples from different classes. Sparse learning has been proven to be a powerful technique in supervised feature selection (Nie et al. 2010; Gu and Han 2011; Tang and Liu 2012a), which enables feature selection to be embedded in the classification (or regression) problem. As most data is unlabeled and it is very expensive to label the data, unsupervised feature selection attracts more and more attentions in recent years (Wolf and Shashua 2005; He et al. 2005; Boutsidis et al. 2009; Yang et al. 2011; Qian and Zhai 2013; Alelyani et al. 2013).

Without label information to define feature relevance, a number of alternative criteria have been proposed for unsupervised feature selection. One commonly used criterion is to select features that can preserve the data similarity or manifold structure constructed from the whole feature space (He et al. 2005; Zhao and Liu 2007). In recent years, applying sparse learning in unsupervised feature selection has attracted increasing attention. These methods usually generate cluster labels via clustering algorithms and then transform unsupervised feature selection into sparse learning based supervised feature selection with these generated cluster labels such as Multi-cluster feature selection (MCFS) (Cai et al. 2010), Nonnegative Discriminative Feature Selection (NDFS) (Li et al. 2012), and Robust Unsupervised Feature Selection (RUFS) (Qian and Zhai 2013).

In this paper, we propose a novel unsupervised feature selection algorithm, i.e., Embedded Unsupervised Feature Selection (EUFS). Unlike existing unsupervised feature selection methods such as MCFS, NDFS or RUFS, which transform unsupervised feature selection into sparse learning based supervised feature selection with cluster labels generated by clustering algorithms, we directly embed feature selection into a clustering algorithm via sparse learning without the transformation (see Figure 1). This work theoretically extends the current state-of-the-art unsupervised feature selection, algorithmically expands the capability of unsupervised feature selection, and empirically demonstrates the efficacy of the new algorithm. The major contributions of this paper are summarized next.

• Providing a way to directly embed unsupervised feature selection algorithm into a clustering algorithm via sparse learning instead of transforming it into sparse learning based supervised feature selection with cluster labels;

• Proposing an embedded feature selection framework EUFS, which selects features in unsupervised scenarios with sparse learning; and
Conducting experiments on various datasets to demonstrate the effectiveness of the proposed framework EUFS.

The rest of this paper is organized as follows. In Section 2, we give details about the embedded unsupervised feature selection framework EUFS. In Section 3, we introduce a method to solve the optimization problem of the proposed framework. In Section 4, we show empirical evaluation with discussion. In section 5, we present the conclusion with future work.

**Embedded Unsupervised Feature Selection**

Throughout this paper, matrices are written as boldface capital letters and vectors are denoted as boldface lowercase letters. For an arbitrary matrix $M \in \mathbb{R}^{m \times n}$, $M_{ij}$ denotes the $(i, j)$-th entry of $M$ while $m_i$ and $m_j$ mean the $i$-th row and $j$-th column of $M$ respectively. $\|M\|_F$ is the Frobenius norm of $M$ and $\text{Tr}(M)$ is the trace of $M$ if $M$ is square. $\langle A, B \rangle$ equals $\text{Tr}(A^T B)$, which is the standard inner product between two matrices. $I$ is the identity matrix and $1$ is a vector whose elements are all $1$. The $l_{2,1}$-norm is defined as $\|M\|_{2,1} = \sum_{i=1}^{m} \|m_i\| = \sum_{i=1}^{m} \sqrt{\sum_{j=1}^{n} M_{ij}^2}$.

Let $X \in \mathbb{R}^{N \times d}$ be the data matrix with each row $x_i \in \mathbb{R}^{1 \times d}$ being a data instance. We use $F = \{f_1, \ldots, f_d\}$ to denote the $d$ features and $f_1, \ldots, f_d$ are the corresponding feature vectors. Assume that each feature has been normalized, i.e., $\|f_j\|_2 = 1$ for $j = 1, \ldots, d$. Suppose that we want to cluster $X$ into $k$ clusters $(C_1, C_2, \ldots, C_k)$ under the matrix factorization framework as:

$$\begin{align*}
\min_{U,V} & \quad \|X - UV^T\|_F^2 \\
\text{s.t.} & \quad U \in \{0, 1\}^{N \times k}, \quad U^T 1 = 1
\end{align*}$$

where $U \in \mathbb{R}^{N \times k}$ is the cluster indicator and $V \in \mathbb{R}^{d \times k}$ is the latent feature matrix. The problem in Eq.(1) is difficult to solve due to the constraint on $U$. Following the common relaxation for label indicator matrix (Von Luxburg 2007; Tang and Liu 2012b), the constraint on $U$ is relaxed to orthogonality, i.e., $U^T U = I$, $U \geq 0$. After the relaxation, Eq.(1) can be rewritten as:

$$\begin{align*}
\min_{U,V} & \quad \|X - UV^T\|_F^2 \\
\text{s.t.} & \quad U^T U = I, \quad U \geq 0
\end{align*}$$

Another significance of the orthogonality constraint on $U$ is to allow us to perform feature selection via $V$, which can be stated by the follow theorem:

**Theorem 1.** Let $X = [f_1, f_2, \ldots, f_d]$, and $\|f_i\| = 1$ for $i = 1, \ldots, d$. We use $UV^T$ to reconstruct $X$, i.e., $\hat{X} = UV^T$. If $U$ is orthogonal, then we can perform feature selection via $V$.

**Proof.** Since $\hat{X} = UV^T$, we have $\hat{f}_i = Uv_i$. Then

$$\|\hat{f}_i\|_2 = \|Uv_i\|_2 = (v_i, U^T U v_i)^{1/2} = \|v_i\|_2 (3)$$

Consider the case that $\|v_i\|_2$ is close to 0, which indicates that the reconstructed feature representation $\|\hat{f}_i\|_2$ is close to 0. $\|\hat{f}_i\|_2 = 1$ means $\hat{f}_i$ is not well reconstructed via $\hat{f}_i$, which suggests that this corresponding feature could not be representative and we should exclude such features to have a better reconstruction. One way to do this is to add a selection matrix $\text{diag}(p)$ to $X$ and $V$ as,

$$\|X \text{diag}(p) - U(\text{diag}(p)V)^T\|_F^2$$

where $p = \{0, 1\}^d$ with $p_i = 1$ if the $i$-th feature is selected and otherwise $p_i = 0$, which completes the proof.

With Theorem 1, if we want to select $m$ features for the clustering algorithm in Eq.(2), we can rewrite it as:

$$\begin{align*}
\min_{U,V} & \quad \|X \text{diag}(p) - U(\text{diag}(p)V)^T\|_F^2 \\
\text{s.t.} & \quad U^T U = I, \quad U \geq 0 \\
p \in \{0, 1\}^d, \quad p^T 1 = m
\end{align*}$$

The constraint on $p$ makes Eq.(5) mixed integer programming (Boyd and Vandenberghe 2004), which is difficult to solve. We relax the problem in the following way. First, the following theorem suggests that we can ignore the selection matrix on $X$ as:

$$\begin{align*}
\min_{U,V} & \quad \|X - U(\text{diag}(p)V)^T\|_F^2 \\
\text{s.t.} & \quad U^T U = I, \quad U \geq 0 \\
p \in \{0, 1\}^d, \quad p^T 1 = m
\end{align*}$$

**Theorem 2.** The optimization problems in Eq.(5) and Eq.(6) are equivalent.

**Proof.** One way to prove Theorem 2 is to show that the objective functions in Eq.(5) and Eq.(6) are equivalent. For Eq.(5), we have

$$\begin{align*}
\|X \text{diag}(p) - U(\text{diag}(p)V)^T\|_F^2
= & \sum_{i=1}^{d} \|p_i f_i - p_i Uv_i^T\|_F^2 \\
= & \sum_{i, p_i = 1} \|f_i - Uv_i^T\|_F^2
\end{align*}$$

471
And for Eq.(6), we have
\[ \| X - U (\text{diag}(p) V)^T \|^2_F = \sum_{i=1}^d \| f_i - p_i U v_i^T \|^2_F \]
\[ = \sum_{i: p_i = 1} \| f_i - U v_i^T \|^2_F + (N - m) \]
which complete the proof.

We observe that diag(p) and V is as the form of diag(p) V in Eq.(6). Since p is a binary vector and N - m rows of the diag(p) are all zeros, diag(p) V is a matrix where elements of many rows are all zeros. This motivates us to absorb the diag(p) into V, i.e., V = diag(p) V, and add l2,1 norm on V to achieve feature selection as
\[ \arg \min_{U, V} \\| X - U V^T \|^2_{2,1} + \alpha \| V \|^2_{2,1} \]
\[ \text{s.t. } U^T U = I, U \succeq 0 \] (9)

To take advantage of information from attribute-value part, i.e., X, similar data instances should have similar labels, according to the spectral analysis (Von Luxburg 2007), we further add the following term to force similar instances with similar labels as:
\[ \min_{W} \text{Tr}(U^T L U) \] (11)
where L = D - S is the Laplacian matrix and D is a diagonal matrix with its elements defined as D_{ii} = \sum_{j=1}^N S_{ij}. S \in \mathbb{R}^{N \times N} denotes the similarity matrix based on X, which is obtained through RBF kernel as
\[ S_{ij} = e^{-\frac{\| x_i - x_j \|^2}{\sigma^2}} \] (12)

Putting Eq.(10) and Eq.(11) together, the proposed framework EUFS is to solve the following optimization problem:
\[ \arg \min_{U, V} \\| X - U V^T \|^2_{2,1} + \alpha \| V \|^2_{2,1} + \beta \text{Tr}(U^T L U) \]
\[ \text{s.t. } U^T U = I, U \succeq 0 \] (13)

**Optimization Algorithm**

The objective function in Eq.(13) is not convex in both U and V but is convex if we update the two variables alternatively. Following (Huang et al. 2014), we use Alternating Direction Method of Multiplier (ADMM) (Boyd et al. 2011) to optimize the objective function. By introducing two auxiliary variables E = X - U V^T and Z = U, we can convert Eq.(13) into the following equivalent problem,
\[ \arg \min_{U, V, E, Z} \| E \|^2_{2,1} + \alpha \| V \|^2_{2,1} + \beta \text{Tr}(Z^T L U) \]
\[ \text{s.t. } E = X - U V^T, Z = U, U^T U = I, Z \succeq 0 \] (14)
which can be solved by the following ADMM problem
\[ \min_{U, V, E, Y_1, Y_2, \mu} \| E \|^2_{2,1} + \alpha \| V \|^2_{2,1} + \beta \text{Tr}(Z^T L U) \]
\[ + \langle Y_1, Z - U \rangle + \langle Y_2, X - U V^T - E \rangle \]
\[ \text{s.t. } U^T U = I, Z \succeq 0 \] (15)
where Y_1, Y_2 are two Lagrangian multipliers and \mu is a scalar to control the penalty for the violation of equality constraints E = X - U V^T and Z = U.

**Update E**

To update E, we fix the other variables except E and remove terms that are irrelevant to E. Then Eq.(15) becomes
\[ \min_{E} \frac{1}{2} \| E - (X - U V^T + \frac{1}{\mu} Y_2) \|^2_F + \frac{1}{\mu} \| E \|^2_{2,1} \] (16)
The equation has a closed form solution by the following Lemma (Liu et al. 2009)

**Lemma 3.** Let Q = [q_1; q_2; \ldots; q_m] be a given matrix and \lambda a positive scalar. If the optimal solution of
\[ \min_{W} \frac{1}{2} \| W - Q \|^2_F + \lambda \| W \|^2_{2,1} \]
is W^*, then the i-th row of W^* is
\[ w^*_i = \begin{cases} (1 - \frac{\lambda}{\| q_i \|}) q_i, & \text{if } \| q_i \| > \lambda \\ 0, & \text{otherwise} \end{cases} \] (18)

Apparently, if we let Q = X - U V^T + \frac{1}{\mu} Y_2, then using Lemma 3, E can be updated as
\[ e_i = \begin{cases} (1 - \frac{1}{\| q_i \|}) q_i, & \text{if } \| q_i \| > \frac{1}{\mu} \\ 0, & \text{otherwise} \end{cases} \] (19)

**Update V**

To update V, we fix the other variables except V and remove terms that are irrelevant to V, then Eq.(15) becomes
\[ \min_{V, U^T = I} \frac{\mu}{2} \| X - U V^T - E + \frac{1}{\mu} Y_2 \|^2_F + \alpha \| V \|^2_{2,1} \] (20)
Using the fact that U^T U = I, we can reformulate Eq.(20) as
\[ \min_{V} \frac{1}{\mu} \| V - (X - E + \frac{1}{\mu} Y_2) U \|^2_F + \frac{\alpha}{\mu} \| V \|^2_{2,1} \] (21)
Again, the above equation has a closed form solution according to Lemma 3. Let K = (X - E + \frac{1}{\mu} Y_2) U, then
\[ v_i = \begin{cases} (1 - \frac{\alpha}{\mu \| k_i \|}) k_i, & \text{if } \| k_i \| > \frac{\alpha}{\mu} \\ 0, & \text{otherwise} \end{cases} \] (22)
Update $Z$

Similarly, to update $Z$, we fix $U, V, E, Y_1, Y_2, \mu$ and remove terms irrelevant to $Z$, then Eq.(15) becomes

$$\min_{Z \succeq 0} \frac{\mu}{2} \|Z - U\|_F^2 + \beta \text{Tr}(Z^T L U) + \langle Y_1, Z - U \rangle$$  \hspace{1cm} (23)

We can rewrite Eq.(23) by putting the second and third terms to the quadratic term and get a compact form

$$\min_{Z \succeq 0} \|Z - T\|_F^2$$  \hspace{1cm} (24)

where $T$ is defined as

$$T = (U - \frac{1}{\mu} Y_1 - \frac{\beta}{\mu} L U)$$  \hspace{1cm} (25)

Eq.(24) can be further decomposed to element-wise optimization problems as

$$\min_{Z_{ij} \geq 0} (Z_{ij} - T_{ij})^2$$  \hspace{1cm} (26)

Clearly, the optimal solution of the above problem is

$$Z_{ij} = \max(T_{ij}, 0)$$  \hspace{1cm} (27)

Update $U$

Optimizing Eq.(15) with respect to $U$ yields the equation

$$\min_{U \in \mathbb{R}^{d \times n}} \frac{\mu}{2} \|X - UV^T - E\|_F^2 + \frac{\mu}{2} \|Z - U\|_F^2 + \beta \text{Tr}(Z^T L U) + \langle Y_2, X - UV^T - E \rangle$$

By expanding Eq.(28) and dropping terms that are independent of $U$, we arrive at

$$\min_{U \in \mathbb{R}^{d \times n}} \frac{\mu}{2} \|U\|_F^2 + \langle N, U \rangle$$

where $N$ is defined as

$$N = \frac{1}{\mu} Y_1 + Z - \beta L Z + (X - E - \frac{1}{\mu} Y_2) V$$

We can further write the above equation into a more compact form as

$$\min_{U \in \mathbb{R}^{d \times n}} \|U - N\|_F^2$$

And now we have converted the objective function of updating $U$ to the classical Orthogonal Procrutes problem (Schönemann 1966) and can be solved using the following lemma (Huang et al. 2014)

**Lemma 4.** Given the objective in Eq.(31), the optimal $U$ is defined as

$$U = PQ^T$$

where $P$ and $Q$ are the left and right singular vectors of the economic singular value decomposition (SVD) of $N$.

Update $Y_1, Y_2$ and $\mu$

After updating the variables, we now need to update the ADMM parameters. According to (Boyd et al. 2011), they are updated as follows

$$Y_1 = Y_1 + \mu(Z - U)$$  \hspace{1cm} (33)

$$Y_2 = Y_2 + \mu(X - UV^T - E)$$  \hspace{1cm} (34)

$$\mu = \max(\rho \mu, \mu_{\text{max}})$$  \hspace{1cm} (35)

Here, $\rho > 1$ is a parameter to control the convergence speed and $\mu_{\text{max}}$ is a larger number to prevent $\mu$ becomes too large.

With these updating rules, EUFS algorithm is summarized in Algorithm 1.

---

**Algorithm 1 Embedded Unsupervised Feature Selection**

**Input:** $X \in \mathbb{R}^{N \times d}$, $\alpha, \beta, n$, latent dimensional $k$

**Output:** $n$ features for the dataset

1. Initialize $\mu = 10^{-3}, \rho = 1.1, \mu_{\text{max}} = 10^{10}, U = 0, V = 0$ (or initialized using K-means)
2. repeat
   3. Calculate $Q = X - UV^T + \frac{1}{\mu} Y_2$
   4. Update $E$
      $$\mathbf{e}_i = \begin{cases} 
      (1 - \frac{1}{\mu} \|\mathbf{q}_i\|) \mathbf{q}_i, & \text{if } \|\mathbf{q}_i\| > \frac{1}{\mu} \\
      0, & \text{otherwise} \end{cases}$$
   5. Calculate $K = (X - E + \frac{1}{\mu} Y_2)^T U$
   6. Update $V$
      $$\mathbf{v}_i = \begin{cases} 
      (1 - \frac{\alpha}{\mu} \|\mathbf{k}_i\|) \mathbf{k}_i, & \text{if } \|\mathbf{k}_i\| > \frac{\alpha}{\mu} \\
      0, & \text{otherwise} \end{cases}$$
   7. Calculate $T$ using Eq.(25)
   8. Update $Z$ using Eq.(27)
   9. Calculate $N$ according to Eq.(30)
   10. Update $U$ by Lemma 4
   11. Update $Y_1, Y_2, \mu$
12. until convergence
   13. Sort each feature of $X$ according to $||\mathbf{v}_i||_2$ in descending order and select the top-$n$ ranked ones

---

**Parameter Initialization**

One way to initialize $U$ and $V$ is to simply set them to be 0. As the algorithm runs, the objective function will gradually converge to the optimal value. To accelerate the convergence speed, following the common way of initializing NMF, we can use k-means to initialize $U$ and $V$. To be specific, we apply k-means to cluster rows of $X$ and get the soft cluster indicator $U$. $V$ is simply set as $X^T U$. $\mu$ is typically set in the range of $10^{-4}$ to $10^{-3}$ initially depending on the datasets and is updated in each iteration. $\mu_{\text{max}}$ is set to be a large value such as $10^{10}$ to give $\mu$ freedom to increase but prevent it from being too large. $\rho$ is empirically set to 1.1 in our algorithm. The larger $\rho$ is , the faster $\mu$ becomes larger and the more we penalize the deviation of the equality constraint, which makes it converges faster. However, we may sacrifice some precision of the final objective function with large $\rho$. 

473
Convergence Analysis

The convergence of our algorithm depends on the convergence of the ADMM. The detailed convergence proof of ADMM can be found in (Goldstein et al. 2012; Boyd et al. 2011). The convergence criteria can be set as $\frac{||J_{t+1} - J_t||}{||J_t||} < \epsilon$, where $J_t$ is the objective function value in Eq.(14) and $\epsilon$ is some tolerance value. In practice, we can control the number of iterations by setting a maximum iteration value. Our experiments find that our algorithm converges within 110 iterations for all the datasets we used.

Time Complexity Analysis

The computation cost for $E$ depends on the computation of $Q = X - UV^T + \frac{1}{\mu}Y_2$ and update of $E$. Since $U$ is sparse, i.e., each row of $U$ only has one nonzero element, then the computation cost is $O(Nd)$ and $O(Nd)$, respectively.

Similiarly, the computation cost for $V$ involves the computation of $K = (X - E + \frac{1}{\mu}Y_2)^T U$ and update of $V$, which is $O(Nd)$ again due to the sparsity of $U$.

The main computation cost for $Z$ is the computation of $T = (U - \frac{1}{\mu}Y_1^T - \frac{1}{\mu}LU)$, which is $O(k^2)$ due to the sparsity of both $U$ and $L$.

The main computation cost of $U$ involves the computation of $N$ and its SVD decomposition, which is $O(Ndk)$ and $O(Nk^2)$. The computational cost for $Y_1$ and $Y_2$ are both $O(Nd)$. Therefore, the overall time complexity is $O(Ndk + Nk^2)$. Since $d \gg k$, the final computation cost if $O(Ndk)$ for each iteration.

Experimental Analysis

In this section, we conduct experiments to evaluate the effectiveness of EUFS. After introducing datasets and experimental settings, we compare EUFS with the state-of-the-art unsupervised feature selection methods. Further experiments are conducted to investigate the effects of important parameters on EUFS.

Datasets

The experiments are conducted on 6 publicly available benchmark datasets, including one Mass Spectrometry (MS) dataset ALLAML (Fodor 1997), two microarray datasets, i.e., Prostate Cancer gene expression (Prostate-GE)

<table>
<thead>
<tr>
<th>Dataset</th>
<th># of Samples</th>
<th># of Features</th>
<th># of Classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALLAML</td>
<td>72</td>
<td>7192</td>
<td>2</td>
</tr>
<tr>
<td>COIL20</td>
<td>1440</td>
<td>1024</td>
<td>20</td>
</tr>
<tr>
<td>PIE10P</td>
<td>210</td>
<td>1024</td>
<td>10</td>
</tr>
<tr>
<td>TOX-171</td>
<td>171</td>
<td>5748</td>
<td>4</td>
</tr>
<tr>
<td>PIX10P</td>
<td>100</td>
<td>10000</td>
<td>10</td>
</tr>
<tr>
<td>Prostate-GE</td>
<td>102</td>
<td>5996</td>
<td>2</td>
</tr>
</tbody>
</table>

Experimental Settings

Following the common way to evaluate unsupervised feature selection algorithms, we assess EUFS in terms of clustering performance (Zhao and Liu 2007; Li et al. 2012). We compare EUFS with the following representative unsupervised feature selection algorithms:

- **All Features:** All original features are adopted
- **LS:** Laplacian Score (He et al. 2005) which evaluates the importance of a feature through its power of locality preservation
- **MCFS:** Multi-Cluster Feature Selection (Cai et al. 2010) which selects features by a joint framework of nonnegative spectral analysis and $l_{2,1}$-norm regularization
- **NDFS:** Nonnegative Discriminative Feature Selection (Li et al. 2012) which selects features by a joint framework of nonnegative spectral analysis and $l_{2,1}$-norm regularization
- **RUFs:** Robust Unsupervised Feature Selection (Qian and Zhai 2013) which jointly performs robust learning and $l_{2,1}$-norm minimization.

Two widely used evaluation metrics, accuracy (ACC) and normalized mutual information (NMI), are employed to evaluate the quality of clusters. The larger ACC and NMI are, the better performance is.

There are some parameters to be set. Following (Qian and Zhai 2013), for LS, MCFS, NDFS, RUFs and EUFS, we fix the neighborhood size to be 5 for all the datasets. To fairly compare different unsupervised feature selection methods, we tune the parameters for all methods by a “grid-search” strategy from $\{10^{-6}, 10^{-4}, \ldots, 10^4, 10^6\}$. For EUFS, we set the latent dimension as the number of clusters. How to determine the optimal number of selected features is still an open problem (Tang and Liu 2012a), we set the number of selected features 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 the experiments 20 times and the average results with standard deviation are reported.

Experimental Results

The experimental results of different methods on the datasets are summarized in Table 2 and Table 3. We make the follow-

---

1The implementation of EUFS can be found from http://www.public.asu.edu/ swang187/
2ALLAML and Prostate-GE are publicly available from https://sites.google.com/site/feipingjie/file
3TOX-171, PIX10P and PIE10P are publicly available from http://featureselection.asu.edu/datasets.php
4COIL20 is publicly available from http://www.cad.zju.edu.cn/home/dengca/Data/MLData.html
Table 2: Clustering results(ACC%±std) of different feature selection algorithms on different datasets. The best results are highlighted in bold. The number in parentheses is the number of features when the performance is achieved.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>ALL Features</th>
<th>Laplacian Score</th>
<th>MCFS</th>
<th>NDFS</th>
<th>RUFs</th>
<th>EUFS</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALLAML</td>
<td>67.3±6.72</td>
<td>73.2±5.52(150)</td>
<td>68.4±10.4(100)</td>
<td>69.4±0.00(100)</td>
<td>72.2±0.00(150)</td>
<td><strong>73.6±0.00(100)</strong></td>
</tr>
<tr>
<td>COIL20</td>
<td>53.6±3.83</td>
<td>55.2±2.84(250)</td>
<td>59.7±4.03(250)</td>
<td>60.1±4.26(300)</td>
<td>62.7±3.51(150)</td>
<td><strong>63.4±5.47(100)</strong></td>
</tr>
<tr>
<td>PIE10P</td>
<td>30.8±2.29</td>
<td>36.0±2.95(150)</td>
<td>44.3±3.20(50)</td>
<td>40.5±4.51(100)</td>
<td>42.6±4.61(50)</td>
<td><strong>46.4±2.69(50)</strong></td>
</tr>
<tr>
<td>TOX-171</td>
<td>41.5±3.88</td>
<td>47.5±3.33(200)</td>
<td>42.5±5.15(100)</td>
<td>46.1±2.55(100)</td>
<td>47.8±3.78(300)</td>
<td><strong>49.5±2.57(100)</strong></td>
</tr>
<tr>
<td>PIX10P</td>
<td>74.3±12.1</td>
<td>76.6±8.10(150)</td>
<td>75.9±8.59(200)</td>
<td>76.7±8.52(200)</td>
<td>73.2±9.40(300)</td>
<td><strong>76.8±5.88(150)</strong></td>
</tr>
<tr>
<td>Prostate-GE</td>
<td>58.1±0.44</td>
<td>57.5±0.49(300)</td>
<td>57.3±0.50(300)</td>
<td>58.3±0.50(100)</td>
<td>59.8±0.00(50)</td>
<td><strong>60.4±0.80(100)</strong></td>
</tr>
</tbody>
</table>

Table 3: Clustering results(NMI%±std) of different feature selection algorithms on different datasets. The best results are highlighted in bold. The number in parentheses is the number of features when the performance is achieved.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>ALL Features</th>
<th>Laplacian Score</th>
<th>MCFS</th>
<th>NDFS</th>
<th>RUFs</th>
<th>EUFS</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALLAML</td>
<td>8.55±5.62</td>
<td>15.0±1.34(100)</td>
<td>11.7±12.2(50)</td>
<td>7.20±0.30(300)</td>
<td>12.0±0.00(150)</td>
<td><strong>15.1±0.00(100)</strong></td>
</tr>
<tr>
<td>COIL20</td>
<td>70.6±1.95</td>
<td>70.3±1.73(300)</td>
<td>72.4±1.90(150)</td>
<td>72.1±1.75(300)</td>
<td>73.1±1.69(150)</td>
<td><strong>77.2±2.75(100)</strong></td>
</tr>
<tr>
<td>PIE10P</td>
<td>32.2±3.47</td>
<td>38.5±1.44(50)</td>
<td><strong>54.3±3.9(50)</strong></td>
<td>46.0±3.14(100)</td>
<td>49.6±5.15(50)</td>
<td>49.8±3.10(150)</td>
</tr>
<tr>
<td>TOX-171</td>
<td>17.8±5.20</td>
<td>30.5±2.70(150)</td>
<td>17.7±6.88(100)</td>
<td>22.3±2.41(300)</td>
<td>28.8±2.71(300)</td>
<td><strong>26.0±2.41(100)</strong></td>
</tr>
<tr>
<td>PIX10P</td>
<td>82.8±6.48</td>
<td>84.3±4.63(150)</td>
<td>85.0±4.95(200)</td>
<td>84.8±4.76(200)</td>
<td>81.1±6.23(300)</td>
<td><strong>85.1±4.30(50)</strong></td>
</tr>
<tr>
<td>Prostate-GE</td>
<td>1.95±0.27</td>
<td>1.59±0.21(300)</td>
<td>1.53±0.21(300)</td>
<td>2.02±0.25(100)</td>
<td>2.86±0.00(50)</td>
<td><strong>3.36±0.48(100)</strong></td>
</tr>
</tbody>
</table>

We also perform parameter analysis for some important parameters of EUFS. Due to space limit, we only report the results on COIL20 in Figure 2. The experimental results show that our method is not very sensitive to $\alpha$ and $\beta$. However, the performance is relatively sensitive to the number of selected features, which is a common problem for many unsupervised feature selection methods.

**Conclusion**

We propose a new unsupervised feature selection approach, EUFS, which directly embeds feature selection into a clustering algorithm via sparse learning. It eliminates the need for transforming unsupervised feature selection into the sparse learning based supervised feature selection with pseudo labels. Nonnegative orthogonality is applied on the cluster indicator to make the problem tractable and ensure that feature selection on latent features has similar effects as on original features. $l_2,1$-norm is applied on the cost function to reduce the effects of the noise introduced by the reconstruction of $X$ and feature selection on $V$. Experimental results on 6 different real world datasets validate the unique contributions of EUFS. Future work is to investigate if EUFS can be extended to dimensionality reduction algorithms.

**Acknowledgments**

This material is based upon work supported by, or in part by, the National Science Foundation (NSF) under grant number.
References


Jin Huang, Feiping Nie, Heng Huang, and Chris Ding. Robust manifold nonnegative matrix factorization. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 8(3):11, 2014.


