The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

Dependence Guided Unsupervised Feature Selection

Jun Guo,[1] **Wenwu Zhu**[1,2]
[1] Tsinghua-Berkeley Shenzhen Institue, Tsinghua University, Shenzhen 518055, China
[2] Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China
eeguojun@outlook.com, wwzhu@tsinghua.edu.cn

Abstract

In the past decade, various sparse learning based unsupervised feature selection methods have been developed. However, most existing studies adopt a two-step strategy. *i.e.*, selecting the top-m features according to a calculated descending order and then performing K-means clustering, resulting in a group of sub-optimal features. To address this problem, we propose a Dependence Guided Unsupervised Feature Selection (DGUFS) method to select features and partition data in a joint manner. Our proposed method enhances the inter-dependence among original data, cluster labels, and selected features. In particular, a projection-free feature selection model is proposed based on $l_{2,0}$-norm equality constraints. We utilize the learned cluster labels to fill in the information gap between original data and selected features. Two dependence guided terms are consequently proposed for our model. More specifically, one term increases the dependence of desired cluster labels on original data, while the other term maximizes the dependence of selected features on cluster labels to guide the process of feature selection. Last but not least, an iterative algorithm based on Alternating Direction Method of Multipliers (ADMM) is designed to solve the constrained minimization problem efficiently. Extensive experiments on different datasets consistently demonstrate that our proposed method significantly outperforms state-of-the-art baselines.

1 Introduction

In many applications, high-dimensional features are often correlated, redundant, or even noisy, which may lead to adverse effects such as heavy computational complexity and poor performance (John, Kohavi, and Pfleger 1994; Liu and Motoda 2007). Therefore, various feature selection methods (Zhao and Liu 2007; He et al. 2012; Chang et al. 2014; Wang et al. 2016; Han and Shen 2016; Li, Tang, and Liu 2017; Cheng, Li, and Liu 2017) are proposed to filter out the unimportant features of high-dimensional data.

In terms of label availability, feature selection can be generally grouped into two major categories, *i.e.*, supervised and unsupervised (Kira and Rendell 1992; Kononenko 1994). Supervised feature selection (Raileanu and Stoffel 2004; Yang et al. 2013; Jian et al. 2016; Fan et al. 2017) aims to select a group of discriminative features with the provided class labels of data which contain the essential discrimination. Supervised feature selection methods have drawn much attention over the past decade. However, the acquisition of label information is laborious and time-consuming, which makes some related tasks more challenging. In contrast, unsupervised feature selection (Law, Figueiredo, and Jain 2004; Boutsidis, Drineas, and Mahoney 2009; Witten and Tibshirani 2010) is desired to explore the properties of unlabeled data in real-world applications.

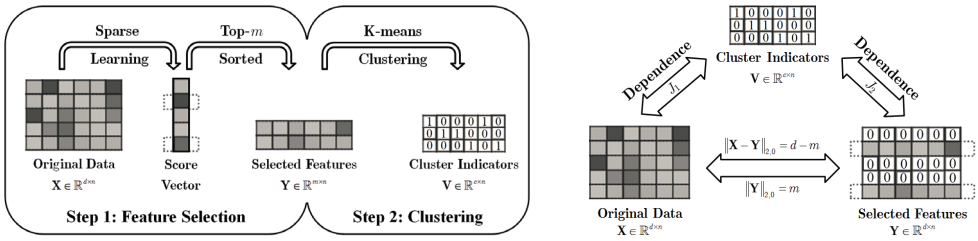
Most unsupervised feature selection methods are based on filters (Dash et al. 2002), wrappers (Roth and Lange 2004), or embedding (Hou et al. 2014; Guo et al. 2017). In the past decade, the success of manifold and sparse learning boosts the research of embedding-based unsupervised feature selection. To achieve encouraging performance, most existing studies on embedding methods introduce pseudo-labels into $l_{2,1}$-norm based sparse learning and emphasize too much on the resulting optimization problem. They usually adopt a two-step strategy. After the overall sparse learning, these unsupervised feature selection methods first calculate the importance for each feature dimension based on some predefined variables, *e.g.*, Laplacian score (He, Cai, and Niyogi 2005), row/column-wise l_2-norm of the latent feature matrix (Wang, Tang, and Liu 2015) or projection matrix (Nie, Zhu, and Li 2016). Next, a score vector can be obtained and then sorted in descending order. Accordingly, previous works select the top-m features to conduct K-means clustering.

However, this two-step strategy for unsupervised feature selection will select a group of sub-optimal features. The main reasons and analyses are two-folds:

- These methods do not directly select m features in the optimization process. Feature selection aims to determine a subset of m features from d $(d > m)$ features, and the selected subset should outperform other $C_d^m - 1$ subsets[1]. The two-step strategy evaluates the importance for all d features, then selects the top-m sorted ones. It is obviously unreasonable to select m features without removing the information of other $d - m$ features in the learning phase.
- The two-step strategy suppresses the inter-dependence among original data, cluster labels, and selected features. From a general perspective, the three aspects are naturally

Copyright © 2018, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

[1] C_d^m denotes the number of m-combinations from a given set of d elements.



(a) Existing sparse learning based unsupervised feature selection methods.

(b) Our proposed DGUFS method.

Figure 1: Most existing sparse learning based unsupervised feature selection methods adopt a two-step strategy, hence they cannot prevent selecting sub-optimal features. In response to this issue, our proposed Dependence Guided Unsupervised Feature Selection (DGUFS) method provides a joint learning framework.

interrelated since they are different carriers for category information. Original data contains all the category information implicitly, so we need to fully exploit it. Cluster labels are the direct form of category information, but they are not given in unsupervised cases. Selected features are the distillation of original data, which should be generated with large inter-cluster differences. Therefore, it is necessary to explore the inter-dependence among them in a joint framework, but not in a two-step way.

This paper makes a beneficial attempt to address these problems. We propose a method called ***D**ependence **G**uided **U**nsupervised **F**eature **S**election* (DGUFS) to select features and partition data in a joint manner. Our method consists of three components. First of all, $l_{2,0}$-norm equality constraints are introduced to present a projection-free feature selection model. Different from previous sparse learning based methods involving $l_{2,1}$-norm, the parameter m (*i.e.*, the number of selected features) is explicitly present in our model. In such a way, an exact number of features can be directly selected. Furthermore, we propose two dependence guided terms. Specifically, one term increases the dependence of desired cluster labels on original data, while the other term maximizes the dependence of selected features on cluster labels to guide the process of feature selection. Based on Alternating Direction Method of Multipliers, we design an effective algorithm to solve the constrained minimization problem. Comparative experimental analysis on several benchmark datasets demonstrates that our proposed method outperforms state-of-the-art sparse learning based unsupervised feature selection methods.

In summary, our main contributions are as follows:

- A joint learning framework for feature selection and clustering is proposed. Our model is projection-free based on $l_{2,0}$-norm equality constraints.
- Two dependence guided terms are consequently designed for our model. Then, original data, cluster labels, and selected features are intimately intertwined.
- An iterative algorithm is designed to efficiently solve the resulting optimization problem. Extensive experiments convincingly demonstrate the superiority of DGUFS.

2 Related Work

In (Cai, Zhang, and He 2010), feature selection preserved the multi-cluster structure of unlabeled data. Yang *et al.* (2011) defined local discriminative scores with an $l_{2,1}$ regularizer. In (Li et al. 2012), feature correlations, local discriminative information, and manifold structures were exploited simultaneously. Qian and Zhai (2013) jointly performed robust learning for both labels and features. In (Wang, Tang, and Liu 2015), feature selection was embedded into sparse learning without projection. By estimating latent cluster centers for the projected data, Han and Kim (2015) conducted simultaneous orthogonal basis clustering and feature selection. Wang *et al.* (2015) simultaneously maximized the total data separability and preserved minimum within-class scatter. In (Zhu et al. 2016), synthesis-analysis dictionary pair was used for unsupervised feature selection. Liu *et al.* (2016) utilized consensus clustering for pseudo-label in feature selection. In (Du and Shen 2015) and (Nie, Zhu, and Li 2016), the similarity matrix was learned adaptively in the joint framework of feature selection and structure learning. Zhu *et al.* (2017) learned an adaptive hypergraph to exploit the structure of unlabeled data when selecting features.

Actually, there are still a lot of works on this topic not mentioned due to 8-page limitation. Despite good performance, these methods cannot prevent selecting sub-optimal features as analyzed in §1.

3 The Proposed DGUFS Method

3.1 Notation and Problem Definition

Except in some specified cases, lowercase letters $(u, \cdots)$ represent scalars. Bold uppercase letters $(\mathbf{U}, \cdots)$ denote matrices, while bold lowercase letters $(\mathbf{u}, \cdots)$ are vectors. $Tr(\mathbf{U})$, $rank(\mathbf{U})$, $\mathbf{U}^{-1}$, and $\mathbf{U}^T$ denote the trace, rank, inverse, and transpose of $\mathbf{U}$, respectively. $\mathbf{U}_{i\cdot}$ presents the i^{th} row of $\mathbf{U}$ and $\mathbf{U}_{\cdot j}$ is the j^{th} column of $\mathbf{U}$. $\mathbf{U}_{ij}$ means the i^{th} element in the j^{th} column of $\mathbf{U}$. $\|\mathbf{U}\|_F$ and $\|\mathbf{U}\|_0$ denote the Frobenius-norm ($\sqrt{\sum_{i,j} \mathbf{U}_{ij}^2}$) and l_0-norm (number of non-zero entries), respectively. $\|\mathbf{U}\|_{2,0}$ is the $l_{2,0}$-norm ($\sum_i \left\| \sqrt{\sum_j \mathbf{U}_{ij}^2} \right\|_0$). $\mathbf{U} \succeq 0$ means that the symmetric matrix

$\mathbf{U}$ is positive semi-definite[2]. $\mathbf{I}$ and $\mathbf{1}$ denote the identity and all-one matrix with compatible sizes, respectively. $diag(\cdot)$ has two meanings: for a vector $\mathbf{u}$, $diag(\mathbf{u})$ returns a diagonal matrix with the elements of $\mathbf{u}$ on the main diagonal; for a square matrix $\mathbf{U}$, $diag(\mathbf{U})$ returns a diagonal matrix with the same main diagonal elements as $\mathbf{U}$.

Let $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_n] \in \mathbb{R}^{d \times n}$ be the original data matrix with n samples from c clusters. $\mathbf{V} = [\mathbf{v}_1, \cdots, \mathbf{v}_n] \in \mathbb{R}^{c \times n}$ denotes the cluster label matrix of $\mathbf{X}$. Each column of $\mathbf{V}$ is a one-hot label vector: $\mathbf{v}_i = [0, 0, \cdots, 1, \cdots, 0, 0]^T$, whose non-zero position indicates the cluster label of $\mathbf{x}_i$. Figure 1 provides a basic example of the original data matrix $\mathbf{X}$ and selected features $\mathbf{Y}$, where $n = 6$, $d = 5$, $m = 2$, and $c = 3$. Samples $\mathbf{x}_1$ and $\mathbf{x}_5$ are from cluster 1, $\mathbf{x}_2$ and $\mathbf{x}_3$ are from cluster 2, $\mathbf{x}_4$ and $\mathbf{x}_6$ are from cluster 3.

Problem Definition: In the joint task of feature selection and clustering, we aim to select $m\,(m < d)$ most discriminative features whose learned pseudo-label indicators are much closer to the true cluster labels. Therefore, the problem can be generally formulated as

$$\begin{aligned} \min_{\mathbf{Y},\mathbf{V}} \quad & J(\mathbf{X}, \mathbf{V}, \mathbf{Y}) \\ s.t. \quad & \mathbf{Y} = diag(\mathbf{s})\,\mathbf{X},\ \mathbf{V} \in \mathbf{\Omega}, \\ & \mathbf{s} \in \{0,1\}^d, \mathbf{s}^T\mathbf{1}_d = m, \end{aligned} \tag{1}$$

where $J(\mathbf{X}, \mathbf{V}, \mathbf{Y})$ is a discrimination promotion function. $\mathbf{\Omega}$ is the candidate set of cluster label matrices that can exactly partition data into c groups.

3.2 $l_{2,0}$-norm Based Projection-free Model

The two constraints imposed on $\mathbf{s}$ make Eq.(1) a mixed integer programming problem which is difficult to solve. Conventional sparse learning based methods employ a projection matrix $\mathbf{P}$ and design a loss term $\ell\left(\mathbf{P}^T\mathbf{Y}, \mathbf{V}\right)$. Consequently, $\mathbf{P}$ and $diag(\mathbf{s})$ are in the form of $\mathbf{P}^T diag(\mathbf{s})$. Note that $\mathbf{s}$ is a binary vector and only $(d-m)$ rows of $diag(\mathbf{s})$ are all zeros. $\mathbf{P}^T diag(\mathbf{s})$ has $(d-m)$ all-zero columns. Most existing unsupervised methods introduce a new matrix $\mathbf{W} = diag(\mathbf{s})\mathbf{P}$ and impose an $l_{2,1}$-norm regularizer or constraint on $\mathbf{W}$ to encourage a relaxed version of Eq.(1). This relaxation leads to a two-step manner, *i.e.*, first using learned variables to calculate a score vector based on $l_{2,1}$-norm, then selecting the top-m sorted features and performing K-means clustering. Therefore, it is sub-optimal for not directly selecting m features in the optimization process.

Different from the above relaxation strategy in previous works, we explicitly utilize $l_{2,0}$-norm equality constraints to propose a projection-free model (2). In such a way, an exact number of features can be directly selected. We rewrite Eq.(1) as[3]

$$\begin{aligned} \min_{\mathbf{Y},\mathbf{V}} \quad & J(\mathbf{X}, \mathbf{V}, \mathbf{Y}) \\ s.t. \quad & \|\mathbf{X} - \mathbf{Y}\|_{2,0} = d - m,\ \|\mathbf{Y}\|_{2,0} = m,\ \mathbf{V} \in \mathbf{\Omega}. \end{aligned} \tag{2}$$

Compared with previous $l_{2,1}$-norm based works, our proposed model (2) is free of sparse projection. The parameter m in our model has an explicit meaning, *i.e.*, the number of selected features. Hence, our method has superiority of selecting an exact number of features.

To enhance the inter-dependence among original data $\mathbf{X}$, learned cluster labels $\mathbf{V}$, and selected features $\mathbf{Y}$, two dependence guided terms are consequently developed for the objective function, *i.e.*, $J(\mathbf{X}, \mathbf{V}, \mathbf{Y}) = \beta J_1(\mathbf{X}, \mathbf{V}) + (1-\beta) J_2(\mathbf{V}, \mathbf{Y})$, where $\beta \in (0,1)$ is a regularization parameter. We put the detailed descriptions of dependence guided terms J_1 and J_2 in the following two subsections.

3.3 Dependence Guided Term J_1

To increase the dependence of desired label matrix $\mathbf{V}$ on the original data $\mathbf{X}$, we design J_1 based on the geometrical structure and discriminative information of data.

Geometrical structure of data: For n original samples $\mathbf{x}_1, \cdots, \mathbf{x}_n$, a similarity matrix $\mathbf{S} \in \mathbb{R}^{n \times n}$ is constructed.

$$\mathbf{S}_{ij} = \begin{cases} 1 & , j \in \mathcal{N}_i \text{ or } i \in \mathcal{N}_j \\ 0 & , \text{otherwise} \end{cases}, \tag{3}$$

where $\mathcal{N}_i$ is a set of indexes indicating the k nearest neighbours of $\mathbf{x}_i$. It is well-known that samples within the same cluster are close to each other while samples from different clusters are far away. $\mathbf{S}_{ij} = 0$ indicates that $\mathbf{x}_i$ and $\mathbf{x}_j$ are completely dissimilar, and thus may have different labels, while $\mathbf{S}_{ij} = 1$ suggests that the two samples are likely to be grouped into the same cluster.

Discriminative information of data: If $\mathbf{x}_j$ belongs to cluster i, $\mathbf{V}_{ij} = 1$ and the remaining entries of the j^{th} column are zeros. $\mathbf{L} \in \mathbb{R}^{n \times n}$ denotes the linear kernel matrix of $\mathbf{V}$, *i.e.*, $\mathbf{L} = \mathbf{V}^T\mathbf{V}$. Thus, if $\mathbf{x}_i$ and $\mathbf{x}_j$ belong to the same cluster, $\mathbf{L}_{ij} = \mathbf{v}_i^T\mathbf{v}_j = 1$; otherwise, $\mathbf{L}_{ij} = \mathbf{v}_i^T\mathbf{v}_j = 0$.

Based on the above two observations, we propose to maximize $Tr\left(\mathbf{S}^T\mathbf{L}\right)$ in order to keep $\mathbf{L}$ as close to the similarity matrix $\mathbf{S}$ as possible. In some related studies, $Tr\left(\mathbf{S}^T\mathbf{L}\right)$ is formulated as $\langle \mathbf{S}, \mathbf{L} \rangle$, named Frobenius inner product. Therefore, $J_1 = -Tr\left(\mathbf{S}^T\mathbf{L}\right)$. Meanwhile, there are some meaningful constraints imposed on $\mathbf{L}$:

- From the preceding, we have $\mathbf{L} = \mathbf{V}^T\mathbf{V}$. Thus, the rank of $\mathbf{L}$ is exactly the number of clusters: $rank(\mathbf{L}) = c$.
- Then, $\mathbf{L}$ is symmetric and positive semi-definite: $\mathbf{L} \succeq 0$.
- Moreover, each element of $\mathbf{L}$ is binary: $\mathbf{L} \in \{0,1\}^{n \times n}$.
- Lastly, $diag(\mathbf{L}) = \mathbf{I}$. This constraint means that a sample cannot be split into different clusters.

3.4 Dependence Guided Term J_2

To maximize the dependence of selected features $\mathbf{Y}$ on the desired label matrix $\mathbf{V}$, we design J_2 based on Hilbert-Schmidt Independence Criterion (HSIC).

HSIC is a kernel-based dependence metric for random variables (Gretton et al. 2005). It measures the dependence between $\mathbf{y}$ and $\mathbf{v}$ by computing the Hilbert-Schmidt-norm of the cross-covariance operator over the domain $\mathcal{Y} \times \mathcal{V}$ in Reproducing Kernel Hilbert Spaces (RKHSs). Suppose that $\mathcal{Q}$ and $\mathcal{U}$ are two RKHSs in $\mathcal{Y}$ and $\mathcal{V}$, respectively. Hence, by Riesz representation theorem, there are two feature mappings $\phi(\mathbf{y}) : \mathcal{Y} \to \mathbb{R}$ and $\psi(\mathbf{v}) : \mathcal{V} \to \mathbb{R}$,

[2]Conventionally, definiteness is not for asymmetric matrices.

[3]In practice, it is often the case that the original data $\mathbf{X}$ does not have all-zero rows. Therefore, Eq.(1) and (2) are equivalent.

such that the kernel function $K(\mathbf{y}, \mathbf{y}')$ returns the inner product $\phi(\mathbf{y})^T \phi(\mathbf{y}')$ in $\mathcal{Q}$, and $L(\mathbf{v}, \mathbf{v}')$ returns the inner product $\psi(\mathbf{v})^T \psi(\mathbf{v}')$ in $\mathcal{U}$. HSIC can be empirically estimated in the RKHSs by a finite number of samples. Let $\{(\mathbf{y}_i, \mathbf{v}_i)\}_{i=1}^n \subseteq \mathcal{Y} \times \mathcal{V}$ denote n observations that are independently and identically drawn from the joint distribution $\Pr_{\mathcal{Y}\times\mathcal{V}}$. Then, HSIC $= \frac{1}{(n-1)^2} Tr(\mathbf{KHLH})$, where $\mathbf{K}, \mathbf{L}, \mathbf{H} \in \mathbb{R}^{n\times n}$. $\mathbf{K}_{ij} = K(\mathbf{y}_i, \mathbf{y}_j)$ and $\mathbf{L}_{ij} = L(\mathbf{v}_i, \mathbf{v}_j)$. $\mathbf{H} = \mathbf{I} - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T$ is the centering matrix[4].

According to (Gretton et al. 2005), maximizing the empirical estimate of HSIC will lead to the maximization of the dependency between two random variables. In accordance with §3.3, we adopt linear kernel matrix, *i.e.*, $\mathbf{K} = \mathbf{Y}^T\mathbf{Y}$ and $\mathbf{L} = \mathbf{V}^T\mathbf{V}$. Therefore, $J_2 = -Tr\left(\mathbf{Y}^T\mathbf{YHV}^T\mathbf{VH}\right)$.

3.5 Overall Model

Taking all together, the overall model of our proposed DGUFS method is written as

$$\begin{array}{ll}\min\limits_{\mathbf{Y},\mathbf{V}} & -\beta Tr\left(\mathbf{S}^T\mathbf{L}\right) - (1-\beta)\, Tr\left(\mathbf{Y}^T\mathbf{YHV}^T\mathbf{VH}\right) \\ s.t. & \|\mathbf{X}-\mathbf{Y}\|_{2,0} = d-m,\ \|\mathbf{Y}\|_{2,0} = m, \\ & \mathbf{V}\in\mathbf{\Omega},\ \mathbf{L} = \mathbf{V}^T\mathbf{V},\ rank(\mathbf{L}) = c, \\ & \mathbf{L}\succeq 0,\ \mathbf{L}\in\{0,1\}^{n\times n},\ diag(\mathbf{L}) = \mathbf{I}. \end{array} \tag{4}$$

The meaningful constraints on $\mathbf{L}$ inspire us to solve $\mathbf{L}$ directly. As for $\mathbf{V}$, we can adopt some classical algorithms to decompose $\mathbf{L}$ at the end of optimization. We postpone the calculation of $\mathbf{V}$ until next section. Since the $rank(\cdot)$ equality constraint makes Eq.(4) difficult to solve, we have converted it into a Lagrange multiplier. Based on these considerations, we finally try to optimize

$$\begin{array}{ll}\min\limits_{\mathbf{Y},\mathbf{L}} & -\beta Tr\left(\mathbf{S}^T\mathbf{L}\right) - (1-\beta)\, Tr\left(\mathbf{Y}^T\mathbf{YHLH}\right) \\ & +\alpha rank(\mathbf{L}) \\ s.t. & \|\mathbf{X}-\mathbf{Y}\|_{2,0} = d-m,\ \|\mathbf{Y}\|_{2,0} = m, \\ & \mathbf{L}\succeq 0,\ \mathbf{L}\in\{0,1\}^{n\times n},\ diag(\mathbf{L}) = \mathbf{I}, \end{array} \tag{5}$$

where $\alpha > 0$ is also a regularization parameter.

4 Optimization

4.1 Optimization Procedure

We can adopt Alternating Direction Method of Multiplier (ADMM) (Boyd et al. 2011) to solve problem (5). By introducing two auxiliary variables $\mathbf{Z}$ and $\mathbf{M}$, Eq.(5) is first rewritten as the following form:

$$\begin{array}{ll}\min\limits_{\mathbf{Y},\mathbf{Z},\mathbf{M},\mathbf{L}} & -\beta Tr\left(\mathbf{S}^T\mathbf{L}\right) - (1-\beta)\, Tr\left(\mathbf{Z}^T\mathbf{YHLH}\right) \\ & +\alpha rank(\mathbf{L}) \\ s.t. & \|\mathbf{X}-\mathbf{Z}\|_{2,0} = d-m,\ \|\mathbf{Y}\|_{2,0} = m,\ \mathbf{Z} = \mathbf{Y}, \\ & \mathbf{L} = \mathbf{M} - diag(\mathbf{M}) + \mathbf{I}, \\ & \mathbf{M}\in\{0,1\}^{n\times n},\ \mathbf{L}\succeq 0. \end{array} \tag{6}$$

The constraint $\mathbf{L} = \mathbf{M} - diag(\mathbf{M}) + \mathbf{I}$ guarantees that $diag(\mathbf{L}) = \mathbf{I}$. Besides, with the binary constraint imposed on $\mathbf{M}$, all elements of $\mathbf{L}$ are binary. Thus, the two constraints $\mathbf{L} = \mathbf{M} - diag(\mathbf{M}) + \mathbf{I}$ and $\mathbf{M}\in\{0,1\}^{n\times n}$ in Eq.(6) are equivalent to the two constraints $\mathbf{L}\in\{0,1\}^{n\times n}$ and $diag(\mathbf{L}) = \mathbf{I}$ in Eq.(5).

The augmented Lagrangian function of Eq.(6) is

$$\begin{array}{ll}\min\limits_{\mathbf{Y},\mathbf{Z},\mathbf{M},\mathbf{L},\mathbf{\Lambda}_1,\mathbf{\Lambda}_2,\mu} & -\beta Tr\left(\mathbf{S}^T\mathbf{L}\right) - (1-\beta)\, Tr\left(\mathbf{Z}^T\mathbf{YHLH}\right) \\ & +\alpha rank(\mathbf{L}) + Tr\left[\mathbf{\Lambda}_1^T(\mathbf{Z}-\mathbf{Y})\right] \\ & +Tr\left[\mathbf{\Lambda}_2^T(\mathbf{L}-\mathbf{M}+diag(\mathbf{M})-\mathbf{I})\right] \\ & +\frac{\mu}{2}\left(\|\mathbf{Z}-\mathbf{Y}\|_F^2 + \|\mathbf{L}-\mathbf{M}+diag(\mathbf{M})-\mathbf{I}\|_F^2\right) \\ s.t. & \|\mathbf{X}-\mathbf{Z}\|_{2,0} = d-m,\ \|\mathbf{Y}\|_{2,0} = m, \\ & \mathbf{M}\in\{0,1\}^{n\times n},\ \mathbf{L}\succeq 0, \end{array} \tag{7}$$

where $\mathbf{\Lambda}_1$ and $\mathbf{\Lambda}_2$ are two Lagrangian multipliers. $\mu > 0$ is a penalty parameter. Eq.(7) can be alternately optimized:

1) **Update $\mathbf{Y}$**: With other variables fixed, we require to solve the following problem:

$$\begin{array}{ll}\min\limits_{\mathbf{Y}} & \left\|\mathbf{Y} - \left(\mathbf{Z} + \frac{(1-\beta)\mathbf{ZHLH}+\mathbf{\Lambda}_1}{\mu}\right)\right\|_F^2 \\ s.t. & \|\mathbf{Y}\|_{2,0} = m. \end{array} \tag{8}$$

It can be efficiently solved by the following Theorem (Luo, Ding, and Huang 2010; Cai, Nie, and Huang 2013).

Theorem 1. *The optimal solution of the optimization problem* $\min_{\mathbf{Y}} \|\mathbf{Y}-\mathbf{U}\|_F^2,\ s.t.\ \|\mathbf{Y}\|_{2,0} = m$ *takes the form* $\mathbf{y}^{\pi(i)} = \begin{cases} \mathbf{u}^{\pi(i)} & , i \le m \\ \mathbf{0} & , i > m \end{cases}$, *where* $\mathbf{y}^{\pi(i)}$ *and* $\mathbf{u}^{\pi(i)}$ *are the* $\pi(i)^{th}$ *rows of* $\mathbf{Y}$ *and* $\mathbf{U}$, *respectively.* π *is the sorting index vector such that* $\left\|\mathbf{u}^{\pi(1)}\right\|_F \ge \left\|\mathbf{u}^{\pi(2)}\right\|_F \ge \cdots \ge \left\|\mathbf{u}^{\pi(d)}\right\|_F$.

Algorithm 1 can be utilized to obtain the optimal solution. Apparently, if we let $\mathbf{U} = \mathbf{Z} + \frac{(1-\beta)\mathbf{ZHLH}+\mathbf{\Lambda}_1}{\mu}$, then $\mathbf{Y}$ can be updated by Algorithm 1.

2) **Update $\mathbf{Z}$**: When other variables are fixed, we solve the following problem:

$$\begin{array}{ll}\min\limits_{\mathbf{Z}} & \left\|(\mathbf{X}-\mathbf{Z}) - \left(\mathbf{X}-\mathbf{Y} - \frac{(1-\beta)\mathbf{YHLH}-\mathbf{\Lambda}_1}{\mu}\right)\right\|_F^2 \\ s.t. & \|\mathbf{X}-\mathbf{Z}\|_{2,0} = d-m. \end{array} \tag{9}$$

The optimal $(\mathbf{X}-\mathbf{Z})$ can also be solved by Algorithm 1 with $\mathbf{U} = \mathbf{X}-\mathbf{Y} - \frac{(1-\beta)\mathbf{YHLH}-\mathbf{\Lambda}_1}{\mu}$. The optimal $\mathbf{Z}$ can be consequently acquired.

3) **Update $\mathbf{M}$**: With other variables fixed, we solve

$$\begin{array}{ll}\min\limits_{\mathbf{M}\in\{0,1\}^{n\times n}} & -Tr\left[\mathbf{\Lambda}_2^T(\mathbf{M}-diag(\mathbf{M})+\mathbf{I}-\mathbf{L})\right] \\ & +\frac{\mu}{2}\|\mathbf{M}-diag(\mathbf{M})+\mathbf{I}-\mathbf{L}\|_F^2. \end{array} \tag{10}$$

According to the optimization rules in (Li, Cheong, and Zhou 2014), we can update $\mathbf{M}$ in two steps. Firstly, we solve $\min_{\mathbf{M}'\in\{0,1\}^{n\times n}} \left\|\mathbf{M}' - \left(\mathbf{L}+\frac{\mathbf{\Lambda}_2}{\mu}\right)\right\|_F^2$. The solution is obtained by thresholding:

$$\mathbf{M}'_{ij} = \begin{cases} 1 & , \text{if } \left(\mathbf{L}+\frac{\mathbf{\Lambda}_2}{\mu}\right)_{ij} \ge \frac{1}{2} \\ 0 & , \text{if } \left(\mathbf{L}+\frac{\mathbf{\Lambda}_2}{\mu}\right)_{ij} < \frac{1}{2} \end{cases}. \tag{11}$$

Secondly, we update $\mathbf{M} = \mathbf{M}' - diag(\mathbf{M}') + \mathbf{I}$.

[4] In the experiments, we absorb $\frac{1}{n-1}$ into $\mathbf{H}$ as $\mathbf{H} = \frac{\mathbf{H}}{n-1}$.

Algorithm 1 Solve the optimization problem in Theorem 1

Input:
The matrix $\mathbf{U}$ with d rows;
The number of desired features m.
Output:
The objective matrix $\mathbf{Y}$.
1: Calculate a vector $\mathbf{f} \in \mathbb{R}^{d\times 1}$, where $\mathbf{f}_i = \sqrt{\sum_j \mathbf{U}_{ij}^2}$;
2: Sort $\mathbf{f}$ in descending order and find out the indexes vector $\mathbf{g} = [g_1, \cdots, g_m]^T$ corresponding to top-m sorted entries;
3: Assign the i^{th} row of $\mathbf{U}$ to the i^{th} row of $\mathbf{Y}$ if $i \in \mathbf{g}$; assign all-zero row vector to the i^{th} row of $\mathbf{Y}$ if $i \notin \mathbf{g}$.

4) **Update $\mathbf{L}$**: Based on the update of $\mathbf{M}$, we update $\mathbf{L}$ by solving the following optimization:

$$\begin{aligned}\min_{\mathbf{L}\succeq 0}\ & -\beta Tr\left(\mathbf{S}^T\mathbf{L}\right) - (1-\beta)\,Tr\left(\mathbf{Z}^T\mathbf{YHLH}\right) \\ & +\alpha rank(\mathbf{L}) + Tr\left[\mathbf{\Lambda}_2^T\left(\mathbf{L}-\mathbf{M}\right)\right] + \tfrac{\mu}{2}\left\|\mathbf{L}-\mathbf{M}\right\|_F^2,\end{aligned} \tag{12}$$

which is equivalent to $\min_{\mathbf{L}\succeq 0}\ \left\|\mathbf{L}-\mathbf{A}\right\|_F^2 + \frac{2\alpha}{\mu} rank(\mathbf{L})$, where $\mathbf{A} = \mathbf{M} + \frac{(1-\beta)\mathbf{HY}^T\mathbf{ZH} + \beta\mathbf{S} - \mathbf{\Lambda}_2}{\mu}$. Using Theorem 2 proposed by (Li, Cheong, and Zhou 2014), $\mathbf{L}$ is updated by setting $\eta = \frac{2\alpha}{\mu}$.

Theorem 2. *For any square matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$, the unique closed-form solution of $\min_{\mathbf{L}\succeq 0}\ \left\|\mathbf{L}-\mathbf{A}\right\|_F^2 + \eta rank(\mathbf{L})$ takes the form $\mathbf{L}^* = \mathbf{Q}\mathcal{T}_\eta(\mathbf{\Omega})\mathbf{Q}^T$, where $\tilde{\mathbf{A}} = \mathbf{Q}\mathbf{\Omega}\mathbf{Q}^T$ is the eigen-decomposition of $\tilde{\mathbf{A}} = \frac{\mathbf{A}+\mathbf{A}^T}{2}$. $\mathcal{T}_\eta(\mathbf{\Omega})$ is an element-wise function acting on the diagonal matrix $\mathbf{\Omega}$, and defined as $\mathcal{T}_\eta(\mathbf{\Omega}_{ii}) = \begin{cases} \mathbf{\Omega}_{ii} & , \mathbf{\Omega}_{ii} > \sqrt{\eta} \\ 0 & , \mathbf{\Omega}_{ii} \le \sqrt{\eta} \end{cases}$.*

5) **Update $\mathbf{\Lambda}_1$, $\mathbf{\Lambda}_2$ and μ**:

$$\mathbf{\Lambda}_1 = \mathbf{\Lambda}_1 + \mu\left(\mathbf{Z}-\mathbf{Y}\right), \tag{13}$$

$$\mathbf{\Lambda}_2 = \mathbf{\Lambda}_2 + \mu\left(\mathbf{L}-\mathbf{M}\right), \tag{14}$$

$$\mu = \min\left(\rho\mu, \mu_{\max}\right), \tag{15}$$

where $\min(\cdot,\cdot)$ returns the minimum value. $\rho > 1$ controls the convergence speed, and $\mu_{\max}$ is a large number to prevent μ from becoming too large.

As summarized in Algorithm 2, these update steps are alternatively performed until convergence.

4.2 Algorithmic Analysis

It is worth noting that although there is no established theory in literature for the global convergence of ADMM applied to non-convex problems as the one solved in this paper. In practice, we set a maximum iteration number.

The computational complexity for $\mathbf{Y}$, $\mathbf{Z}$ and $\mathbf{\Lambda}_1$ are the same $\mathcal{O}(nd)$, while the computational complexity for $\mathbf{M}$ and $\mathbf{\Lambda}_2$ are both $\mathcal{O}\left(n^2\right)$. The time complexity for $\mathbf{L}$ is $\mathcal{O}(n^3)$, which involves the eigen-decomposition. Therefore, the total computational complexity of Algorithm 2 is $\mathcal{O}(n^3 + nd)$ for each iteration. The overall time cost tends to be small, because we find that Algorithm 2 converges within less than 50 iterations for all datasets in our experiments.

Here are the time complexities of other previous algorithms: MCFS (Cai, Zhang, and He 2010): $\mathcal{O}(dn^2 + cm^3 + cnm^2 + d\log d)$, UDFS (Yang et al. 2011): $\mathcal{O}(d^3)$, NDFS (Li et al. 2012): $\mathcal{O}(cn + d^3)$, and RUFS (Qian and Zhai 2013): $\mathcal{O}(cn^2 + d^3)$, EUFS (Wang, Tang, and Liu 2015): $\mathcal{O}(dcn + nc^2)$. Thus, our DGUFS method has an acceptable time cost. To handle data with a large n, it is reasonable for our method to adopt a sample-level hierarchical strategy, which is a promising future work.

Algorithm 2 The proposed DGUFS method

Input:
Data matrix $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_n] \in \mathbb{R}^{d\times n}$;
Number of desired features m and latent clusters c;
Number of each sample's nearest neighbours k;
Regularization parameters β and α.
Output:
Selected features $\mathbf{Y}$ and cluster labels $\mathbf{V}$.
1: Compute the similarity matrix $\mathbf{S}$ via Eq.(3);
2: Initialize $\mathbf{Y} = \mathbf{Z} = \mathbf{\Lambda}_1 = \mathbf{0}_{d\times n}$, $\mathbf{L} = \mathbf{M} = \mathbf{\Lambda}_2 = \mathbf{0}_{n\times n}$, $\rho = 1.1$, $\mu_{\max} = 10^{10}$, and $\mu = 10^{-6}$.
3: **repeat**
4: Update $\mathbf{Y}$ by utilizing Algorithm 1 to solve (8);
5: Update $\mathbf{Z}$ by utilizing Algorithm 1 to solve (9);
6: Update $\mathbf{M}$ via (11) followed by $\mathbf{M} = \mathbf{M}' - diag\left(\mathbf{M}'\right) + \mathbf{I}$;
7: Update $\mathbf{L}$ according to Theorem 2;
8: Update $\mathbf{\Lambda}_1$, $\mathbf{\Lambda}_2$, and μ via (13), (14) and (15);
9: **until** convergence
10: Compute $\mathbf{V}$ by decomposing $\mathbf{L}$.

4.3 Discussion

Calculation for the cluster labels: Readers can adopt the Constrained Boolean Matrix Factorization (CBMF) algorithm proposed by (Li, Cheong, and Zhou 2014). Here, we recommend a simple yet effective strategy. Different from the time-consuming CBMF method, we can employ eigen decomposition and exhaustive search.

$\mathbf{L} = \mathbf{R}diag\left(\xi\right)\mathbf{R}^T$ is the eigen decomposition of $\mathbf{L}$, which is equivalent to

$$\mathbf{L} = \left[\mathbf{R}diag\left(\sqrt{\xi}\right)\right]\left[\mathbf{R}diag\left(\sqrt{\xi}\right)\right]^T, \tag{16}$$

where ξ is a vector storing the n non-negative eigenvalues of the positive semi-definite $\mathbf{L} = \mathbf{V}^T\mathbf{V}$. Then, we can regard $\hat{\mathbf{V}} \triangleq \left[\mathbf{R}diag\left(\sqrt{\xi}\right)\right]^T$ as the approximation of $\mathbf{V}$. The final cluster labels can be obtained by an exhaustive search, *i.e.*, determining the position of the largest element (in magnitude) in each column of $\hat{\mathbf{V}}$.

Relaxation for the auxiliary variable $\mathbf{M}$: Motivated by (Li, Cheong, and Zhou 2014), we recommend adding an l_0 penalty on $\mathbf{M}$ to enforce sparsity on its entries and avoid the trivial solution. Besides, to make the problem tractable, readers can relax the constraint $\mathbf{M} \in \{0,1\}^{n\times n}$ to obtain real valued entries $\mathbf{M} \in [0,1]^{n\times n}$. Then, the only change lies in Eq.(11), which has a new formulation.

Adding a regularization $\gamma\left\|\mathbf{M}\right\|_0$ will introduce an extra parameter γ. We have observed that the above relaxation with $\gamma = 0.005$ can help our proposed method achieve better performance and converge faster.

Table 1: Dataset Description.

Dataset	# of Samples	# of Features	# of Classes
ALLAML	72	7129	2
Prostate-GE	102	5966	2
LUNG	203	3312	5
UMIST	575	644	20
PIX 10P	100	10000	10
PIE 10P	210	2420	10

5 Experiments

5.1 Experimental Settings

In this section, we compare our proposed DGUFS approach with state-of-the-art methods on several benchmark datasets.

Datasets: one mass spectrometry dataset ALLAML (Fodor 1997), one microarray dataset Prostate-GE (Singh et al. 2002), one cancer dataset LUNG (Bhattacharjee et al. 2001), and three face image datasets UMIST (Graham and Allinson 1998), PIX 10P[5], and PIE 10P (Gross et al. 2008). Detailed information is listed in Table 1.

Comparing algorithms are as follows:

- **All Features:** All original features are adopted as the baseline in the experiments.
- **Laplacian Score (LapScore):** Features corresponding to the largest Laplacian scores are selected to preserve the local manifold structure well (He, Cai, and Niyogi 2005).
- **Multi-Cluster Feature Selection (MCFS):** Features are selected based on sparse regression and spectral analysis problem (Cai, Zhang, and He 2010).
- **Unsupervised Discriminative Feature Selection (UDFS):** Features are selected by joint $l_{2,1}$-norm minimization and discriminative analysis (Yang et al. 2011).
- **Nonnegative Discriminative Feature Selection (NDFS):** Features are selected by joint $l_{2,1}$-norm regularized regression and nonnegative spectral analysis (Li et al. 2012).
- **Robust Unsupervised Feature Selection (RUFS):** Features are selected by joint $l_{2,1}$-norm regularized regression and $l_{2,1}$-norm based Nonnegative Matrix Factorization (NMF) with local learning (Qian and Zhai 2013).
- **Embedded Unsupervised Feature Selection (EUFS):** Features are selected by joint $l_{2,1}$-norm minimization and graph embedding (Wang, Tang, and Liu 2015).

Settings: Different parameters may be utilized for different datasets. For fair comparison, we tune the parameters for all unsupervised feature selection algorithms by grid-search strategy. Meanwhile, there are some parameters to be set in advance. Parameter k is set to 5 for all datasets to specify the size of neighbourhood. We set the numbers of selected features as $\{50, 100, \cdots, 300\}$ for all datasets. For the NDFS method (Li et al. 2012), we fix $\gamma = 10^8$ to guarantee the orthogonality. Then, we report the best results from the optimal parameters for all methods. Some results in the tables are from the published papers. Following the experiment settings in the previous works, two commonly used evaluation metrics, *i.e.*, Accuracy (ACC) and Normalized Mutual Information (NMI) are employed to measure the performance in clustering. The larger ACC and NMI are, the better performance is. At the **L**-step of our method, iteration based eigen-decomposition uses a random start, which is varied. For other comparative methods, the initialization of K-means is varied. Therefore, we repeat all experiments 20 times with random initialization. The mean and standard deviation (STD) of ACC and NMI for all algorithms are reported.

[5]http://peipa.essex.ac.uk/ipa/pix/faces/

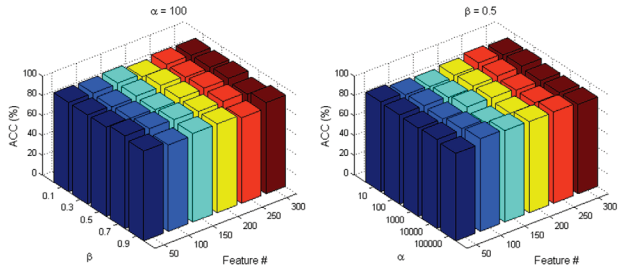


Figure 2: Accuracy (ACC) over PIX 10P dataset with different β, α, and selected feature numbers.

5.2 Clustering with Selected Features

We report the comparison results of clustering in Table 2 and 3. The number in the parentheses denotes the number of selected features when the performance is achieved. We have three **observations**. Firstly, it is necessary for clustering tasks to conduct feature selection. The selected features can not only reduce the computational cost, but also improve the clustering performance. Secondly, simultaneous feature selection and clustering can achieve better performance than using two-step strategies, *i.e.*, clustering after feature selection. Thirdly, our proposed DGUFS method tends to achieve better results with usually fewer selected features.

As shown in Table 2 and 3, DGUFS outperforms its competitors on all datasets. The **major reasons** are two-folds: On the one hand, our work is free of sparse projection. The parameter m in our model has an explicit meaning, *i.e.*, the number of selected features. Hence, our model has superiority of selecting an exact number of features in the optimization process. On the other hand, our method enhances the inter-dependence among original data, cluster labels, and selected features in a joint learning framework. One dependence guided term strengthens the dependence of desired cluster labels on original data, while the other dependence guided term maximizes the dependence of selected features on cluster labels to guide the learning process.

5.3 In-depth Empirical Study of DGUFS

In this subsection, we study the sensitivity of parameters. After checking all details, we can find that the numbers of hyper-parameters for our competitors MCFS, UDFS, NDFS, RUFS, and EUFS are 4, 4, 6, 5, and 5, respectively. However, not all of them are major parameters to be fine-tuned.

Table 2: Clustering results (ACC%±STD). The best results are in boldface.

	ALLAML	Prostate-GE	LUNG	UMIST	PIX 10P	PIE 10P
All Features	67.3±6.72	58.1±0.44	72.0±8.88	42.1±2.3	74.3±12.1	30.8±2.29
LapScore	73.2±5.52(150)	57.5±0.49(300)	62.1±9.05(300)	45.1±3.42(200)	76.6±8.10(150)	36.0±2.95(100)
MCFS	68.4±10.4(100)	57.3±0.50(300)	66.3±7.91(300)	45.4±3.21(150)	75.9±8.59(200)	44.3±3.20(50)
UDFS	70.4±0.41(150)	57.7±0.45(300)	68.2±7.84(300)	45.3±2.74(300)	75.8±8.11(250)	41.6±3.82(100)
NDFS	69.4±0.00(100)	58.3±0.50(100)	68.9±9.06(300)	48.2±3.62(150)	76.7±8.52(200)	40.5±4.51(100)
RUFS	72.2±0.00(150)	59.8±0.00(50)	70.4±8.28(250)	49.1±3.25(100)	73.2±9.40(300)	42.6±4.61(50)
EUFS	73.6±0.00(100)	60.4±0.80(100)	72.5±8.57(300)	51.5±3.09(150)	76.8±5.88(150)	46.4±2.69(50)
DGUFS	**78.4±1.31**(200)	**65.3±1.11**(250)	**79.2±7.14**(100)	**57.1±3.36**(100)	**82.1±4.98**(100)	**51.9±2.04**(50)

Table 3: Clustering results (NMI%±STD). The best results are in boldface.

	ALLAML	Prostate-GE	LUNG	UMIST	PIX 10P	PIE 10P
All Features	8.55±5.62	1.95±0.27	51.8±5.42	63.9±2.5	82.8±6.48	32.2±3.47
LapScore	15.0±1.34(100)	1.59±0.21(300)	44.9±5.12(300)	65.9±1.94(250)	84.3±4.63(150)	38.5±1.44(50)
MCFS	11.7±10.2(50)	1.53±0.21(300)	47.7±4.48(300)	67.3±2.61(150)	85.0±4.95(200)	54.3±3.39(50)
UDFS	12.7±0.42(150)	1.55±0.20(300)	51.6±5.08(300)	65.2±1.62(300)	84.9±4.87(250)	47.3±3.02(50)
NDFS	7.20±0.30(300)	2.02±0.25(100)	50.3±5.21(250)	66.5±2.20(200)	84.8±4.76(200)	46.0±3.14(100)
RUFS	12.0±0.00(150)	2.86±0.00(50)	51.1±5.11(250)	68.8±2.39(150)	81.1±6.23(300)	49.6±5.15(50)
EUFS	15.1±0.00(100)	3.36±0.48(100)	53.0±4.98(250)	69.7±1.78(150)	85.1±4.30(50)	49.8±3.10(150)
DGUFS	**19.1±1.78**(150)	**6.59±0.84**(250)	**60.2±4.69**(100)	**74.4±1.67**(100)	**89.2±3.26**(50)	**55.0±2.86**(50)

Hence, these papers do not need to list all hyper-parameters in their algorithms. Parameters for minor cases, such as determining convergence tolerance and avoiding singularity or zero-denominator, can be set to small values, *e.g.*, 10^{-6}. The number of desired features m should be determined by users. In comparison experiments, all methods can have the same range of m, *e.g.*, $m = \{50, 100, \cdots, 300\}$ for fairness. The number of latent clusters c is given prior, which is common in existing works. The number of neighbouring parameter k is set to 5 for all datasets to specify the size of neighbourhood. This setting is consistent to previous works, *e.g.*, MCFS, NDFS, RUFS, and EUFS. The remaining parameters are main parameters for each method, which should be fine-tuned. Actually, if there is no constraint, all methods will tune them in a range of $\{10^{-6}, 10^{-4}, \cdots, 10^{6}\}$.

As aforementioned, the number of neighboring parameter k is set to 5 for all datasets to specify the size of neighborhood, which is consistent with the settings in previous works. There are two major parameters to tune in our algorithm, *i.e.*, β and α. Since $\beta \in (0, 1)$ in our method, it cannot be fine-tuned in the same range of $\{10^{-6}, 10^{-4}, \cdots, 10^{6}\}$ as previous works. We set β from 0.1 to 0.9 with 0.2 as interval. Generally, $c \ll n$. The rank of $\mathbf{L} \in \mathbb{R}^{n\times n}$, *i.e.*, $rank(\mathbf{L}) = c$ is usually very small. Hence, we set the corresponding parameter α larger as $\{10^{1}, 10^{2}, \cdots, 10^{5}\}$. Due to the page limit, we only report the results of ACC over PIX 10P dataset. The experiment results are shown in Figure 2. From the two 3D graphs in Figure 2, we can see that the two parameters of DGUFS have relatively wide ranges. Similar trends can be observed on other datasets as well.

Finally, we display some toy examples of our proposed DGUFS method. We randomly select two samples from different classes of the UMIST dataset as toy examples. After conducting our proposed DGUFS method on original images, we select $m = \{50, 100, \cdots, 300\}$ features. For illustration, the selected features are set to green and the unselected features maintain their original values. We draw them in Table 4. As can be seen, the selected features of our proposed method are concentrated. With each fixed number of selected features, our method tends to catch compact and discriminative parts, *e.g.*, hair, eyes, and mouth, which could describe each person's character.

Table 4: Toy examples of DGUFS on UMIST dataset.

Original Image	# of Selected Features					
	50	100	150	200	250	300
						
						

6 Conclusion

In this paper, we have proposed a joint learning framework for feature selection and clustering. A projection-free feature selection model is proposed based on $l_{2,0}$-norm equality constraints. Meanwhile, we explicitly present two dependence guided terms, enhancing the dependence among original data, cluster labels, and selected features. Based on the Alternating Direction Method of Multipliers (ADMM), an iterative algorithm has been designed for efficient optimization. Extensive experiments show that our proposed DGUFS approach outperforms state-of-the-art sparse learning based unsupervised feature selection methods.

Acknowledgments

This work is supported by National Program on Key Basic Research Project No. 2015CB352300 and National Natural Science Foundation of China Major Project No. U1611461.

References

Bhattacharjee, A.; Richards, W. G.; Staunton, J.; Li, C.; Monti, S.; Vasa, P.; Ladd, C.; Beheshti, J.; Bueno, R.; Gillette, M.; et al. 2001. Classification of human lung carcinomas by mrna expression profiling reveals distinct adenocarcinoma subclasses. *National Academy of Sciences (NAS)* 98(24):13790–13795.

Boutsidis, C.; Drineas, P.; and Mahoney, M. W. 2009. Unsupervised feature selection for the k-means clustering problem. In *NIPS*, 153–161.

Boyd, S.; Parikh, N.; Chu, E.; and Peleato, B. 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning* 1(3):1–122.

Cai, X.; Nie, F.; and Huang, H. 2013. Exact top-k feature selection via $l_{2;0}$-norm constraint. In *IJCAI*, 1240–1246.

Cai, D.; Zhang, C.; and He, X. 2010. Unsupervised feature selection for multi-cluster data. In *SIGKDD*, 333–342.

Chang, X.; Nie, F.; Yang, Y.; and Huang, H. 2014. A convex formulation for semi-supervised multi-label feature selection. In *AAAI*, 1171–1177.

Cheng, K.; Li, J.; and Liu, H. 2017. Unsupervised feature selection in signed social networks. In *SIGKDD*, 777–786.

Dash, M.; Choi, K.; Scheuermann, P.; and Liu, H. 2002. Feature selection for clustering − a filter solution. In *ICDM*, 115–122.

Du, L., and Shen, Y. D. 2015. Unsupervised feature selection with adaptive structure learning. In *SIGKDD*, 209–218.

Fan, M.; Chang, X.; Zhang, X.; Wang, D.; and Du, L. 2017. Top-k supervise feature selection via ADMM for integer programming. In *IJCAI*, 1646–1653.

Fodor, S. P. A. 1997. Massively parallel genomics. *Science* 277(5324):393.

Graham, D. B., and Allinson, N. M. 1998. Characterising virtual eigensignatures for general purpose face recognition. *Face Recognition* 446–456.

Gretton, A.; Bousquet, O.; Smola, A.; and Scholkopf, B. 2005. Measuring statistical dependence with hilbert-schmidt norms. In *ALT*, 63–77.

Gross, R.; Matthews, I.; Cohn, J.; Kanade, T.; and Baker, S. 2008. Multi-PIE. In *IEEE International Conference on Automatic Face and Gesture Recognition*, 1–8.

Guo, J.; Guo, Y.; Kong, X.; and He, R. 2017. Unsupervised feature selection with ordinal locality. In *ICME*, 1213–1218.

Han, D., and Kim, J. 2015. Unsupervised simultaneous orthogonal basis clustering feature selection. In *CVPR*, 5016–5023.

Han, Y., and Shen, Y. 2016. Partially supervised graph embedding for positive unlabelled feature selection. In *IJCAI*, 1548–1554.

He, R.; Tan, T.; Wang, L.; and Zheng, W. 2012. $l_{2,1}$ regularized correntropy for robust feature selection. In *CVPR*, 2504–2511.

He, X.; Cai, D.; and Niyogi, P. 2005. Laplacian score for feature selection. In *NIPS*, 507–514.

Hou, C.; Nie, F.; Li, X.; Yi, D.; and Wu, Y. 2014. Joint embedding learning and sparse regression: A framework for unsupervised feature selection. *TCYB* 44(6):793–804.

Jian, L.; Li, J.; Shu, K.; and Liu, H. 2016. Multi-label informed feature selection. In *IJCAI*, 1627–1633.

John, G. H.; Kohavi, R.; and Pfleger, K. 1994. Irrelevant features and the subset selection problem. In *ICML*, 121–129.

Kira, K., and Rendell, L. A. 1992. A practical approach to feature selection. In *International Workshop Machine Learning*, 249–256.

Kononenko, I. 1994. Estimating attributes: analysis and extensions of relief. In *ECML*, 171–182.

Law, M. H. C.; Figueiredo, M. A. T.; and Jain, A. K. 2004. Simultaneous feature selection and clustering using mixture models. *TPAMI* 26(9):1154–1166.

Li, Z.; Yang, Y.; Liu, J.; Zhou, X.; and Lu, H. 2012. Unsupervised feature selection using nonnegative spectral analysis. In *AAAI*, 1026–1032.

Li, Z.; Cheong, L. F.; and Zhou, S. Z. 2014. Scams: Simultaneous clustering and model selection. In *CVPR*, 264–271.

Li, J.; Tang, J.; and Liu, H. 2017. Reconstruction-based unsupervised feature selection: An embedded approach. In *IJCAI*, 2159–2165.

Liu, H., and Motoda, H. 2007. Computational methods of feature selection. Technical report, CRC Press.

Liu, H.; Shao, M.; and Fu, Y. 2016. Consensus guided unsupervised feature selection. In *AAAI*, 1874–1880.

Luo, D.; Ding, C.; and Huang, H. 2010. Towards structural sparsity: An explicit l_2/l_0 approach. In *ICDM*, 344–353.

Nie, F.; Zhu, W.; and Li, X. 2016. Unsupervised feature selection with structured graph optimization. In *AAAI*, 1302–1308.

Qian, M., and Zhai, C. 2013. Robust unsupervised feature selection. In *IJCAI*, 1621–1627.

Raileanu, L. E., and Stoffel, K. 2004. Theoretical comparison between the gini index and information gain criteria. *Annals of Mathematics and Artificial Intelligence* 41(1):77–93.

Roth, V., and Lange, T. 2004. Feature selection in clustering problems. In *NIPS*, 473–480.

Singh, D.; Febbo, P. G.; Ross, K.; Jackson, D. G.; Manola, J.; Ladd, C.; Tamayo, P.; Renshaw, A. A.; DAmico, A. V.; Richie, J. P.; et al. 2002. Gene expression correlates of clinical prostate cancer behavior. *Cancer cell* 1(2):203–209.

Wang, S.; Nie, F.; Chang, X.; Yao, L.; Li, X.; and Sheng, Q. Z. 2015. Unsupervised feature analysis with class margin optimization. In *ECML/PKDD*, 383–398.

Wang, K.; He, R.; Wang, L.; Wang, W.; and Tan, T. 2016. Joint feature selection and subspace learning for cross-modal retrieval. *TPAMI* 38(10):2010–2023.

Wang, S.; Tang, J.; and Liu, H. 2015. Embedded unsupervised feature selection. In *AAAI*, 470–476.

Witten, D. M., and Tibshirani, R. 2010. A framework for feature selection in clustering. *Journal of the American Statistical Association* 105(490):713–726.

Yang, Y.; Shen, H.; Ma, Z.; Huang, Z.; and Zhou, X. 2011. $l_{2,1}$-norm regularized discriminative feature selection for unsupervised learning. In *IJCAI*, volume 22, 1589–1594.

Yang, Y.; Ma, Z.; Hauptmann, A. G.; and Sebe, N. 2013. Feature selection for multimedia analysis by sharing information among multiple tasks. *TMM* 15(3):661–669.

Zhao, Z., and Liu, H. 2007. Spectral feature selection for supervised and unsupervised learning. In *ICML*, 1151–1157.

Zhu, P.; Hu, Q.; Zhang, C.; and Zuo, W. 2016. Coupled dictionary learning for unsupervised feature selection. In *AAAI*, 2422–2428.

Zhu, X.; Zhu, Y.; Zhang, S.; Hu, R.; and He, W. 2017. Adaptive hypergraph learning for unsupervised feature selection. In *IJCAI*, 3581–3587.