Balanced Spectral Feature Selection

Peng Zhou, *Member, IEEE*, Jiangyong Chen, Liang Du, *Member, IEEE*, and Xuejun Li, *Member, IEEE*

***Abstract*—In many real-world unsupervised learning applications, given data with balanced distribution, that is, there are an approximately equal number of instances in each class, we often need to construct a model to reveal such balance. However, in many data, especially the high-dimensional ones, the data in the original feature space often do not present such balance due to the redundant and noisy features. To tackle this problem, we apply an unsupervised spectral feature selection method to select some informative features, which can better reveal the balanced structure of data. Although spectral feature selection is one of the most popular unsupervised feature selection methods and has been widely studied, none of the existing spectral feature selection methods consider the balance property of data. To address this issue, in this article, we propose a novel balanced spectral feature selection (BSFS) method, which not only selects the discriminative features but also picks those to reveal the balanced structure of data. To the best of our knowledge, this is the first spectral feature selection method considering balance structure of data. By introducing a balanced regularization term, we integrate the balanced spectral clustering and feature selection into a unified framework seamlessly. At last, the experiments on benchmark datasets show that the proposed one outperforms the conventional feature selection methods in both clustering performance and balance, which demonstrates the effectiveness and efficiency of the proposed method.**

***Index Terms*—Balanced clustering, feature selection, unsupervised learning.**

I. Introduction

In MANY real-world data mining applications, we often need to discover or reveal the balanced structure for given data with balanced distribution, that is, in each class or cluster, there are an approximately equal number of instances. For example, in wireless sensor networks, the energy load balance should be considered, because the unbalanced structure may lead to the unbalance of energy consumption and may further shorten the network lifetime [1]. Another scenario which needs balanced structure is the cloud computing, where the multiple tenant placement should be balanced for reducing the energy consumption and delay [2]. Therefore, discovering the balanced structure of data is an essential and important task in some data mining applications.

However, although some data are with balanced distribution, it is still difficult to discover their balanced structure in the original feature space. As we know, the original features of data are often noisy and redundant, which makes the balanced structure not so obvious. This problem may be more severe in high-dimensional data. To tackle this problem, we apply feature selection methods to select some informative features to characterize the balanced structure. Feature selection is a fundamental problem in machine learning and has been widely studied [3]–[8]. These methods often select the discriminative and less redundant features for classification or clustering. For example, Nie *et al.* [3] applied $\ell_{2,1}$ regression to pick features leading to a robust feature selection method; Tang *et al.* [9] learned the latent representation for feature selection.

Among the above-mentioned methods, spectral feature selection is one of the most famous unsupervised methods and have been demonstrated promising performance in clustering. However, none of the existing spectral feature selection methods consider the balanced structure of data. This problem may be even more severe in unsupervised learning, because there are no labels that can help us to select features. To address this issue, in this article, we propose a novel balanced spectral feature selection (BSFS) method, which pays attention to the balanced structure of data. To reveal the balanced structure, we apply spectral clustering [10] to generate the pseudolabels and further obtain the clustering structure to guide the feature selection. Unlike traditional spectral clustering, we introduce a balanced term to guarantee the balance of the result. Then, we use the balanced result to guide us to select features. When selecting features, we directly use the $\ell_{2,0}$-norm to pick the exact top-k features without approximations such as $\ell_{2,1}$-norm or $\ell_{2,p}$-norm. To make the balanced clustering and the feature selection be boosted by each other, we seamlessly integrate them into a unified framework.

Since we directly use the $\ell_{2,0}$-norm to select features and involve the balanced term, the optimization of the objective function could be complicated. To tackle this problem, we provide an effective and efficient alternating direction method of multipliers (ADMMs) [11] to iteratively optimize the objective function. We further propose a speedup strategy to reduce the time complexity. Finally, we compare the proposed algorithm with several state-of-the-art unsupervised feature selection methods on nine benchmark datasets. The experimental results demonstrate the effectiveness and efficiency of our proposed one. It is worthy to clarify that one of our assumption is that the given data have a balanced intrinsic structure such that by

Manuscript received 1 August 2021; revised 4 January 2022; accepted 15 March 2022. Date of publication 25 March 2022; date of current version 16 June 2023. This work was supported in part by the National Natural Science Fund of China under Grant 62176001, Grant 61806003, Grant 61922088, and Grant 61972001. This article was recommended by Associate Editor D. Wang. *(Corresponding author: Peng Zhou.)*

Peng Zhou was with the State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing 100190, China. He is now with the School of Computer Science and Technology, Anhui University, Hefei 230601, China (e-mail: zhoupeng@ahu.edu.cn).

Jiangyong Chen and Xuejun Li are with the School of Computer Science and Technology, Anhui University, Hefei 230601, China (e-mail: e18301199@stu.ahu.edu.cn; xjli@ahu.edu.cn).

Liang Du is with the School of Computer and Information Technology, Shanxi University, Taiyuan 030006, China (e-mail: duliang@sxu.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCYB.2022.3160244.

Digital Object Identifier 10.1109/TCYB.2022.3160244

imposing the balanced term, we can select the features that can reveal such balanced structure. The proposed method may be inappropriate to handle the imbalanced data. If the data does not have such balanced structure, since the proposed method still imposes the balanced term, it may not discover the true intrinsic structure that is imbalanced.

The main contributions of this article are as follows.

1) We apply feature selection to discover the balanced structure of data. Different from some existing balanced clustering methods [12]–[14], which are designed for particular clustering methods, our feature selection method provides a more general way to discover the balanced structure, that is, our method can be followed by any standard data analyzing method, like k-means, to show the balanced structure. Moreover, our experiments show that our strategy may achieve a better tradeoff between the clustering performance and balance.
2) We propose a novel spectral feature selection method. Different from the existing methods that only focus on the discriminative features, our method can select not only the discriminative features but also those that can reveal the balanced structure of data. To the best of our knowledge, this is the first spectral feature selection method considering the balanced structure of data.
3) We apply a simple yet effective way to control the balance and integrate the balanced spectral clustering and feature selection into a unified framework and propose an effective and efficient ADMM algorithm to select features and do clustering jointly.
4) The experimental results show that our method outperforms other state-of-the-art unsupervised feature selection methods not only on the clustering performance but also on the balance of results.

The remaining parts of this article are organized as follows. Section II provides some related work. Section III introduces our BSFS in detail. Section IV shows the experimental results. Section V concludes this article and discusses some future work.

II. Related Work

In this section, we briefly introduce some related work of feature selection and balanced clustering.

A. Feature Selection

Feature selection aims to identify and select the informative features in a dataset for some subsequent data analysis methods. According to the accessibility of data labels, it can be roughly classified into three categories: 1) supervised [15], [16]; 2) unsupervised [17]–[19]; and 3) semisupervised [20], [21] methods. Because of the absence of supervised information (i.e., labels), the unsupervised one is more challenging and attracts much attention.

In unsupervised learning scenarios, since no labels can be used, the methods aim to select some informative features to preserve the intrinsic structure of data, such as the graph structure and subspace structure. For example, Zhu *et al.* [22] used the co-regularization technique to make sure that the selected features can preserve data distribution and reconstruction; Feng and Duarte [23] proposed an autoencoder-based feature selection method, which can preserve the broad and local structure of data; Xie *et al.* [24] provided a feature selection method to preserve the distribution of the data; Ye *et al.* [25] tried to preserve the intermediate representation by selected features. Guo and Zhu [26] applied the dependence to guide the feature selection; Shen *et al.* [27] proposed a robust feature selection method to reconstruct the original data that could handle missing data. Du *et al.* [28] selected features to exploit the combination effect of the features.

Since the graph structure is one of the most widely studied structures in machine learning tasks, many feature selection methods, especially the spectral feature selection methods, pick the features to preserve the graph structure. For example, Du and Shen [29] designed an adaptive graph structure and proposed a spectral feature selection method to preserve it; Shang *et al.* [30], [31] constructed a graph with non-negative low-dimensional embedding for spectral feature selection; Zhu *et al.* [32] proposed a spectral feature selection method to characterize the global and local structure of data; Dong *et al.* [33] extended adaptive graph learning to multiview feature selection; Tang *et al.* [34] proposed a robust feature selection method to preserve the graph structure by ℓ_1 norm-based graph regularized term; Zhou *et al.* [35] ensembled multiple graphs to simultaneously learn the consensus graph and selected features; Nie *et al.* [36] learned an optimized graph to select the top-k features; Tang *et al.* [9], [37] applied manifold regularization to select features, and extended these graph base methods to multiview data and obtained a consensus learning guided multiview feature selection method [38]. Besides these conventional graph-based methods, some methods aim to preserve the hypergraph structure. For example, Zhang and Hancock [39] constructed a hypergraph on features that could preserve the high-order relationship of features and then used the hypergraph partition method to choose the features; recently, they further provided an adaptive hypergraph learning method that could simultaneously select features and construct the hypergraph [40]; Luo *et al.* [41] learned features with spatial-spectral hypergraph discriminant analysis to handle the hyperspectral images; Luo *et al.* [42] further constructed an interclass hypergraph and an intraclass hypergraph to guide the feature learning; when handling hyperspectral images, Duan *et al.* [43] proposed a local constrain-based hypergraph learning method for feature learning.

Some methods focused on the subspace structure. For example, Wen *et al.* [44] jointly learned sparse subspace form data and selected features to preserve such subspace structure; Zhu *et al.* [45] selected features for subspace clustering; Fan *et al.* [46], [47] applied the subspace clustering to select features that can preserve both the hard and soft structure; Zheng *et al.* [48] picked the features to preserve the low-rank subspace structure; Tang *et al.* [37], [49] applied the dual Laplacian regularization to learn the selective feature projection, which mapped the data into a low-rank subspace; Li *et al.* [50] learned a feature dictionary subspace for robust and compact feature selection; Zhong and Pun [51] simultaneously did subspace clustering and feature selection.

Although we do not have any true labels of data, we can generate pseudolabels to guide the feature selection. For example, Du *et al.* [52] applied global and local information to jointly select features and learn the pseudolabels; Hou *et al.* [53] selected features to preserve the pseudolabels with manifold embedding; Wang *et al.* [54] proposed a matrix factorization method to preserve the pseudolabels for feature selection; Zhang *et al.* [55] applied the uncorrelated regression model to unsupervised feature selection; Chen *et al.* [56] applied extreme learning machine to generate the pseudolabels and use them to select features. Tang *et al.* [57] used the pseudolabels to preserve the locality and extended it to the multiview data.

Although the above-mentioned methods preserve different structures, they all aim to select the discriminative features. Unlike these methods, we try to select the features that not only are discriminative but also can reveal the balanced structure. Note that compared with our previous balanced method [58], the proposed one has two advantages. On the one hand, [58] used k-means to generate pseudolabels, which may be inappropriate to handle the nonlinear data. In contrast, the proposed method is a spectral-based method that can effectively tackle this problem. On the other hand, when [58] handled the discrete constraints, the time complexity was cubic with the number of features. However, our method can solve the problem more efficiently, and the time complexity is just linear with the dimensions of features.

B. Balanced Clustering

Given a dataset with balanced distribution, balanced clustering aims to put the data into several clusters and make the numbers of data in different clusters remain almost the same. It can be roughly classified into two classes: 1) hard-balanced clustering and 2) soft-balanced clustering.

In the first class, that is, the hard-balanced clustering, the number of the instances in each cluster is strictly set as a fixed number. For example, in [59] and [60], k-means clustering methods that strictly constrain the cluster sizes are proposed; Costa *et al.* [61] proposed a minimum sum-of-squares clustering method for balanced clustering.

Different from hard-balanced clustering, soft-balanced clustering does not require the absolute balance. For example, in [62] and [63], soft-balanced clustering methods are proposed by introducing a penalty regularized term to achieve balance; Liu *et al.* [13] designed an exclusive lasso term to make the clustering results of least square regression be balanced; Li *et al.* [14] applied the exclusive lasso to k-means and min-cut methods to obtain the balance clustering.

In this article, we focus on the soft-balanced clustering because it is more flexible than hard-balanced clustering. Different from these balanced clustering methods, which are designed for particular clustering methods, our algorithm is a feature selection one, which focuses on selecting the informative features to reveal the balance structure. After selecting the features, we can use any standard data analysis methods such as k-means to show a balanced structure. Thus, this article provides a more general way to reveal the balanced structure of data.

III. Balanced Spectral Feature Selection

In this section, we introduce our BSFS method in detail. We first introduce some notations. We use a bold uppercase and lowercase character to denote a matrix and a vector, respectively. For an arbitrary matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$, A_{ij} denotes its (i, j)th element.

A. Formulation

In conventional supervised feature selection methods, given original data with n instances and d features $\mathbf{X} \in \mathbb{R}^{n\times d}$ belonging to c classes and its label matrix $\mathbf{Y} \in \{0, 1\}^{n\times c}$ where if the ith instance belongs to the jth class, then $Y_{ij} = 1$ and other $Y_{im} = 0$ with $m \neq j$, they often learn a row sparse projection matrix $\mathbf{W} \in \mathbb{R}^{d\times c}$ that projects $\mathbf{X}$ into label space. More formally, we can optimize the following formula:

$$\begin{aligned}\min_{\mathbf{W}} \quad & \|\mathbf{Y} - \mathbf{XW}\|_F^2 \\ \text{s.t.} \quad & \|\mathbf{W}\|_{2,0} = k\end{aligned} \tag{1}$$

where $\|\mathbf{W}\|_{2,0}$ is the $\ell_{2,0}$-norm, which counts the number of nonzero rows in $\mathbf{W}$. Since we aim to select k features, we impose the $\ell_{2,0}$-norm constraint on $\mathbf{W}$ to make sure that there are exact k nonzero rows in $\mathbf{W}$. Note that different from many conventional feature selection methods, which applies the $\ell_{2,1}$-norm or $\ell_{2,p}$-norm to approximate $\ell_{2,0}$-norm, we directly use the $\ell_{2,0}$-norm constraint to select the exact top-k features without approximations, which is more desirable in feature selection task.

However, in unsupervised learning, we do not know $\mathbf{Y}$ in advance, which makes the problem more challenging. A natural idea is that we can use some clustering methods to generate the pseudolabels $\mathbf{Y}$. Since spectral clustering [10] is one of the most popular clustering methods and demonstrated promising performance, we use it to generate the pseudolabels. In more detail, we first construct a sparse similarity matrix $\mathbf{S} \in \mathbb{R}^{n\times n}$, where $S_{ij} \geq 0$ is the similarity of the ith and the jth instances. Then, we obtain the normalized Laplacian matrix $\mathbf{L} \in \mathbb{R}^{n\times n}$ by $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{S}\mathbf{D}^{-1/2}$, where $\mathbf{I}$ is an identity matrix and $\mathbf{D} \in \mathbb{R}^{n\times n}$ is a diagonal matrix whose diagonal elements are $D_{ii} = \sum_{j=1}^{n} S_{ij}$. Spectral clustering aims to learn an orthogonal embedding $\mathbf{Y} \in \mathbb{R}^{n\times c}$ of $\mathbf{L}$ by optimizing

$$\begin{aligned}\min_{\mathbf{Y}} \quad & \operatorname{tr}\left(\mathbf{Y}^T\mathbf{L}\mathbf{Y}\right) \\ \text{s.t.} \quad & \mathbf{Y}^T\mathbf{Y} = \mathbf{I}.\end{aligned} \tag{2}$$

Note that in the traditional spectral clustering, it relaxes the label matrix into a real-valued orthogonal matrix $\mathbf{Y}$. In our work, since we need the pseudolabel matrix, we replace the orthogonal constraint back to the 0, 1-constraint without any relaxation, that is, we minimize the following formula:

$$\begin{aligned}\min_{\mathbf{Y}} \quad & \operatorname{tr}\left(\mathbf{Y}^T\mathbf{L}\mathbf{Y}\right) \\ \text{s.t.} \quad & \mathbf{Y} \in \{0, 1\}^{n\times c}, \quad \sum_{j=1}^{c} Y_{ij} = 1\end{aligned} \tag{3}$$

where $\sum_{j=1}^{c} Y_{ij} = 1$ makes sure that each instance just belongs to one cluster. Taking the definition of $\mathbf{L}$ into (3), we have

$$\begin{aligned}\mathrm{tr}\left(\mathbf{Y}^T\mathbf{L}\mathbf{Y}\right) &= \mathrm{tr}\left(\mathbf{Y}^T\mathbf{Y}\right) - \mathrm{tr}\left(\mathbf{Y}^T\mathbf{D}^{-1/2}\mathbf{S}\mathbf{D}^{-1/2}\mathbf{Y}\right)\\ &= n - \mathrm{tr}\left(\mathbf{Y}^T\mathbf{D}^{-1/2}\mathbf{S}\mathbf{D}^{-1/2}\mathbf{Y}\right). \quad (4)\end{aligned}$$

Let $\tilde{\mathbf{S}} = \mathbf{D}^{-1/2}\mathbf{S}\mathbf{D}^{-1/2}$, from (4), we have min $\mathrm{tr}(\mathbf{Y}^T\mathbf{L}\mathbf{Y})$ is equivalent to max $tr(\mathbf{Y}^T\tilde{\mathbf{S}}\mathbf{Y})$. Combining (1) and (3), we obtain the following multiobjective optimization problem:

$$\begin{cases}\min_{\mathbf{W}} & \|\mathbf{Y}-\mathbf{X}\mathbf{W}\|_F^2,\\ \max_{\mathbf{Y}} & \mathrm{tr}\left(\mathbf{Y}^T\tilde{\mathbf{S}}\mathbf{Y}\right).\end{cases}$$

To optimize this multiobjective optimization problem without introducing any more hyperparameters, we rewrite it as the following form:

$$\begin{aligned}\min_{\mathbf{Y},\mathbf{W}} \quad & \frac{\|\mathbf{Y}-\mathbf{X}\mathbf{W}\|_F^2}{\mathrm{tr}\left(\mathbf{Y}^T\tilde{\mathbf{S}}\mathbf{Y}\right)}\\ \text{s.t.} \quad & \|\mathbf{W}\|_{2,0} = k\\ & \mathbf{Y} \in \{0,1\}^{n\times c}, \quad \sum_{j=1}^{c} Y_{ij} = 1. \quad (5)\end{aligned}$$

Note that $\mathbf{Y}$ plays the role as the pseudolabels, so we can achieve a balanced clustering result by imposing some constraints on $\mathbf{Y}$. More specifically, we find that the number of 1's in each column is equal to the number of instances in each cluster. Therefore, by summing $\mathbf{Y}$ by columns, we can obtain the number of instances in every cluster. We denote the number of instances in the jth cluster by $n_j = \sum_{i=1}^{n} Y_{ij}$. Furthermore, since $\sum_{j=1}^{c} n_j = n$, we can obtain the empirical distribution $\mathbf{p} \in \mathbb{R}^c$ of the number of instances in each cluster by $p_j = (n_j/n)$. Since we wish the distribution $\mathbf{p}$ should be as balanced as possible, we can achieve this by maximizing the Shannon entropy of the distribution $\mathbf{p}$, that is, $\max_{\mathbf{p}} -\sum_{j=1}^{c} p_j \log(p_j)$, which is equivalent to $\min_{\mathbf{p}} \sum_{j=1}^{c} p_j \log(p_j)$. Take this regularized term to (5), leading to our BSFS

$$\begin{aligned}\min_{\mathbf{Y},\mathbf{W}} \quad & \frac{\|\mathbf{Y}-\mathbf{X}\mathbf{W}\|_F^2}{\mathrm{tr}\left(\mathbf{Y}^T\tilde{\mathbf{S}}\mathbf{Y}\right)} + \gamma \sum_{j=1}^{c} p_j \log\left(p_j\right)\\ \text{s.t.} \quad & \|\mathbf{W}\|_{2,0} = k, \quad \mathbf{Y} \in \{0,1\}^{n\times c}, \quad \sum_{j=1}^{c} Y_{ij} = 1\\ & p_j = \frac{\sum_{i=1}^{n} Y_{ij}}{n} \quad (6)\end{aligned}$$

where γ is a hyperparameter to control the balance of the clustering result. From (6), we can find that our method has the following two characteristics.

1) The feature selection and balanced clustering are seamlessly integrated into a unified framework. With the selected features, we can achieve a better clustering result. With the balanced regularized term, we can obtain a more balanced result and this result may guide us to select more informative features in turn.
2) We apply $\ell_{2,0}$-norm to directly select the exact top-k features without any approximations and postprocessing such as ranking the features.

B. Optimization

Since (6) includes $\ell_{2,0}$-norm constraint, which is hard to optimize, we apply the ADMM [11] to handle it. More specifically, we introduce an auxiliary variable $\mathbf{V}$ and rewrite (6) as the following:

$$\begin{aligned}\min_{\mathbf{Y},\mathbf{W}} \quad & \frac{\|\mathbf{Y}-\mathbf{X}\mathbf{W}\|_F^2}{\mathrm{tr}\left(\mathbf{Y}^T\tilde{\mathbf{S}}\mathbf{Y}\right)} + \gamma \sum_{j=1}^{c} p_j \log\left(p_j\right)\\ \text{s.t.} \quad & \|\mathbf{V}\|_{2,0} = k, \quad \mathbf{W} = \mathbf{V}\\ & \mathbf{Y} \in \{0,1\}^{n\times c}, \quad \sum_{j=1}^{c} Y_{ij} = 1\\ & p_j = \frac{\sum_{i=1}^{n} Y_{ij}}{n}. \quad (7)\end{aligned}$$

Then, by introducing the Lagrange multipliers $\boldsymbol{\Lambda} \in \mathbb{R}^{d\times c}$ and $\boldsymbol{\rho} \in \mathbb{R}^c$, we obtain its augmented Lagrange function

$$\begin{aligned}\mathcal{L} = & \frac{\|\mathbf{Y}-\mathbf{X}\mathbf{W}\|_F^2}{\mathrm{tr}\left(\mathbf{Y}^T\tilde{\mathbf{S}}\mathbf{Y}\right)} + \gamma \sum_{j=1}^{c} p_j \log\left(p_j\right)\\ & + \mathrm{tr}\left(\boldsymbol{\Lambda}^T(\mathbf{W}-\mathbf{V})\right) + \sum_{j=1}^{c} \rho_j \left(p_j - \frac{\sum_{i=1}^{n} Y_{ij}}{n}\right)\\ & + \frac{\mu}{2}\left(\|\mathbf{W}-\mathbf{V}\|_F^2 + \sum_{j=1}^{c}\left(p_j - \frac{\sum_{i=1}^{n} Y_{ij}}{n}\right)^2\right) \quad (8)\end{aligned}$$

where $\mu > 0$ is an adaptive parameter. Now, we iteratively optimize one variable with other variable fixed.

1) Optimizing $\mathbf{W}$*:* When fixing the other variables, we denote $a = [1/(\mathrm{tr}(\mathbf{Y}^T\tilde{\mathbf{S}}\mathbf{Y}))]$ for simplicity. Then, we obtain the subproblem with $\mathbf{W}$

$$\min_{\mathbf{W}} \; a\|\mathbf{Y}-\mathbf{X}\mathbf{W}\|_F^2 + \mathrm{tr}\left(\boldsymbol{\Lambda}^T(\mathbf{W}-\mathbf{V})\right) + \frac{\mu}{2}\|\mathbf{W}-\mathbf{V}\|_F^2. \quad (9)$$

Since it is an unconstraint quadratic programming, its closed-form solution can be easily obtained by setting its partial derivative w.r.t. $\mathbf{W}$ to zero

$$\mathbf{W} = \left(\mathbf{X}^T\mathbf{X} + \frac{\mu}{2a}\mathbf{I}\right)^{-1}\left(\mathbf{X}^T\mathbf{Y} - \frac{\boldsymbol{\Lambda} - \mu\mathbf{V}}{2a}\right). \quad (10)$$

Unfortunately, the size of $\mathbf{X}^T\mathbf{X} + (\mu/2a)\mathbf{I}$ is $d\times d$, and the time complexity of its inverse is $O(d^3)$. It is inappropriate in feature selection tasks, because in feature selection tasks it often happens that $d \gg n$. To address this issue, we propose a speedup strategy. Note that $\mathbf{X}^T\mathbf{X}$ is constant in the entire learning processing, so we can compute its singular value decomposition (SVD) in advance. When $d > n$, we use $O(n^2d)$ time to compute the SVD of $\mathbf{X}$ as $\mathbf{X} = \mathbf{P}\boldsymbol{\Sigma}\mathbf{Q}^T$ with $\mathbf{P} \in \mathbb{R}^{n\times n}$, $\boldsymbol{\Sigma} \in \mathbb{R}^{n\times n}$, and $\mathbf{Q} \in \mathbb{R}^{d\times n}$, where $\mathbf{P}$ and $\mathbf{Q}$ are orthogonal and $\boldsymbol{\Sigma}$ is diagonal. Note that this SVD can be computed in advance and just needs to be computed once. Then, the SVD of $\mathbf{X}^T\mathbf{X}$ is $\mathbf{X}^T\mathbf{X} = \mathbf{Q}\boldsymbol{\Sigma}^2\mathbf{Q}^T$. According to the Woodbury identity, we have

$$\begin{aligned}\left(\mathbf{X}^T\mathbf{X} + \frac{\mu}{2a}\mathbf{I}\right)^{-1} &= \left(\mathbf{Q}\boldsymbol{\Sigma}^2\mathbf{Q}^T + \frac{\mu}{2a}\mathbf{I}\right)^{-1}\\ &= \frac{2a}{\mu}\mathbf{I} - \frac{4a^2}{\mu^2}\mathbf{Q}\left(\boldsymbol{\Sigma}^{-2} + \frac{2a}{\mu}\mathbf{Q}^T\mathbf{Q}\right)^{-1}\mathbf{Q}^T\\ &= \frac{2a}{\mu}\mathbf{I} - \frac{4a^2}{\mu^2}\mathbf{Q}\left(\boldsymbol{\Sigma}^{-2} + \frac{2a}{\mu}\mathbf{I}\right)^{-1}\mathbf{Q}^T.\end{aligned}$$

Note that $\boldsymbol{\Sigma}^{-2} + (2a/\mu)\mathbf{I}$ is a diagonal matrix and its inverse can be computed in $O(n)$ time. Then, taking it back to (10), we have

$$\begin{aligned}\mathbf{W} &= \left(\frac{2a}{\mu}\mathbf{I} - \frac{4a^2}{\mu^2}\mathbf{Q}\left(\boldsymbol{\Sigma}^{-2} + \frac{2a}{\mu}\mathbf{I}\right)^{-1}\mathbf{Q}^T\right)\left(\mathbf{X}^T\mathbf{Y} - \frac{\boldsymbol{\Lambda} - \mu\mathbf{V}}{2a}\right)\\ &= \frac{1}{\mu}\left(2a\mathbf{X}^T\mathbf{Y} - \boldsymbol{\Lambda} + \mu\mathbf{V}\right)\\ &\quad - \frac{4a^2}{\mu^2}\mathbf{Q}\left(\boldsymbol{\Sigma}^{-2} + \frac{2a}{\mu}\mathbf{I}\right)^{-1}\left(\mathbf{Q}^T\left(\mathbf{X}^T\mathbf{Y} - \frac{\boldsymbol{\Lambda} - \mu\mathbf{V}}{2a}\right)\right).\end{aligned} \tag{11}$$

Therefore, in each iteration, (11) can be computed in $O(ndc)$ time, which is linear with d.

In the case $n > d$, we compute the SVD of $\mathbf{X}^T\mathbf{X}$ in $O(nd^2)$ time, and in each iteration, we compute (10) directly in $O(ndc + d^2c)$, which is linear with n.

2) Optimizing $\mathbf{V}$*:* When optimizing $\mathbf{V}$, we aim to solve the following subproblem:

$$\begin{aligned}&\min_{\mathbf{V}} \left\|\mathbf{V} - \left(\mathbf{W} + \frac{\boldsymbol{\Lambda}}{\mu}\right)\right\|_F^2\\ &\text{s.t. } \|\mathbf{V}\|_{2,0} = k.\end{aligned} \tag{12}$$

Equation (12) also has a closed-form solution. We first let $\mathbf{V} = \mathbf{W} + (\boldsymbol{\Lambda}/\mu)$, and then we look for the k rows with the largest ℓ_2-norm and set other rows to zeros.

3) Optimizing $\mathbf{p}$*:* When optimizing $\mathbf{p}$, we find that this can be decoupled into c independent subproblems. Consider the jth one, which has the following form:

$$\min_{p_j} \mathcal{J} = \gamma p_j \log(p_j) + \rho_j p_j + \frac{\mu}{2}p_j^2 - \mu p_j b_j \tag{13}$$

where $b_j = [(\sum_{i=1}^n Y_{ij})/n]$. Setting its partial derivative w.r.t. p_m to zero, we obtain the following formula:

$$\frac{\partial\mathcal{J}}{\partial p_j} = \gamma\log(p_j) + \mu p_j + \rho_j + \gamma - \mu b_j = 0. \tag{14}$$

It can be solved by any standard root finding algorithms. In our implementation, we solve it by *fzero* function provided in MATLAB.

4) Optimizing $\mathbf{Y}$*:* When fixing other variables, we can rewrite (8) as

$$\begin{aligned}\min_{\mathbf{Y}} &\frac{\|\mathbf{Y} - \mathbf{X}\mathbf{W}\|_F^2}{\mathrm{tr}\left(\mathbf{Y}^T\tilde{\mathbf{S}}\mathbf{Y}\right)} + \sum_{j=1}^{c}\rho_j\left(p_j - \frac{\sum_{i=1}^n Y_{ij}}{n}\right)\\ &+ \frac{\mu}{2}\sum_{j=1}^{c}\left(p_j - \frac{\sum_{i=1}^n Y_{ij}}{n}\right)^2\\ \text{s.t. } &\mathbf{Y} \in \{0, 1\}^{n\times c}, \quad \sum_{j=1}^{c} Y_{ij} = 1.\end{aligned} \tag{15}$$

Algorithm 1 BSFS

```
Input: Data matrix $\mathbf{X}$, normalized graph matrix $\tilde{\mathbf{S}}$, parameter
    $\gamma$ and number of selected features $k$.
Output: Selected features.
Initialize $\mathbf{Y}$ by $\max \mathbf{Y}^T\tilde{\mathbf{S}}\mathbf{Y}$. Compute the SVD of $\mathbf{X}^T\mathbf{X}$.
    Initialize $\mathbf{W}$ by $\min \|\mathbf{Y} - \mathbf{X}\mathbf{W}\|_F^2$, initialize $\mathbf{V} = \mathbf{W}$, $p_j = \sum_{i=1}^n Y_{ij}/n$, $\boldsymbol{\Lambda} = \mathbf{0}$, $\boldsymbol{\rho} = \mathbf{0}$ and $\mu = 1$.
while not converge do
    Compute $\mathbf{W}$ by Eq.(11).
    Compute $\mathbf{V}$ by solving Eq.(12).
    Compute $\mathbf{p}$ by solving Eq.(14).
    Compute $\mathbf{Y}$ by solving Eq.(15).
    Update the Lagrange multipliers by Eq.(16).
end while
Select the $k$ features corresponding to the $k$ non-zero rows
    of $\mathbf{W}$.
```

Note that the constraints guarantee that there is only one 1 in each row of $\mathbf{Y}$. So for simplicity, we can optimize $\mathbf{Y}$ row by row. When optimizing the ith row while fixing the other rows, we define $\hat{\mathbf{y}}_1, \ldots, \hat{\mathbf{y}}_c$ where $\hat{\mathbf{y}}_j \in \{0, 1\}^{1\times c}$ and only the jth element in $\hat{\mathbf{y}}_j$ is 1 and other elements are 0 s. Then, we set the ith row of $\mathbf{Y}$ as $\hat{\mathbf{y}}_1, \ldots, \hat{\mathbf{y}}_c$, respectively, and find the one who leads to the minimum of objective function, and set the ith row of $\mathbf{Y}$ as it.

5) Updating Lagrange Multipliers: We update the Lagrange Multipliers as follows:

$$\begin{cases}\boldsymbol{\Lambda} \leftarrow \boldsymbol{\Lambda} + \mu(\mathbf{W} - \mathbf{V})\\ \rho_j \leftarrow \rho_j + \mu\left(p_j - \frac{\sum_{i=1}^n Y_{ij}}{n}\right)\\ \mu \leftarrow 1.1 * \mu.\end{cases} \tag{16}$$

The entire algorithm is summarized in Algorithm 1. In each subproblem, the objective function decreases monotonically. However, since the objective function (6) is not jointly convex, it is difficult to analyze the convergence of ADMM theoretically. In practice, our method often converges very fast.

C. Time Complexity

As introduced before, if $d > n$, computing the SVD of $\mathbf{X}^T\mathbf{X}$ costs $O(n^2d)$ time. Then, in ADMM algorithms, optimizing $\mathbf{W}$ costs $O(ndc)$ time. The time complexity of optimizing $\mathbf{V}$ is $O(kd + cd)$. When optimizing $\mathbf{p}$, supposing the time complexity of root finding algorithm is $O(t)$, we need $O(ct)$ time to compute $\mathbf{p}$. Since $\tilde{\mathbf{S}}$ is often a sparse graph matrix, supposing in each row there are $O(\kappa)$ nonzero elements in $\tilde{\mathbf{S}}$, computing $tr(\mathbf{Y}^T\tilde{\mathbf{S}}\mathbf{Y})$ costs $O(c\kappa n)$ time. Updating $\mathbf{Y}$ costs $O(ndc + c\kappa n)$ time. Supposing the number of iterations is l, the entire time complexity is $O(n^2d + l(ndc + kd + ct + c\kappa n))$, which is linear with d.

If $d < n$, computing the SVD of $\mathbf{X}^T\mathbf{X}$ costs $O(d^2n)$ time. Updating $\mathbf{W}$ costs $O(ndc + d^2c)$ time. The other parts are similar to the case $d > n$. Therefore, the time complexity is $O(nd^2 + l(d^2c + ndc + kd + ct + c\kappa n))$, which is linear with n.

TABLE I
DESCRIPTION OF THE DATASETS

	#instances	#features	#classes	Type
PIE	1428	1024	68	Image
News4b	3874	5652	4	Text
CBCL	2000	361	2	Image
Basehock	1993	4862	2	Text
USPS49	1673	256	2	Image
Lung	73	325	7	Biology
Leukemia	72	7070	2	Biology
Gisette	7000	5000	2	Biology
MNIST	60000	784	10	Image

Note that compared with some previous methods [58], [64], [65] whose complexity is cubic in d. Our proposed method reduced the time complexity significantly, especially in the case of handling high-dimensional data.

IV. EXPERIMENTS

In this section, to demonstrate the effectiveness and efficiency of the proposed BSFS, we compare it with the state-of-the-art unsupervised feature selection methods on some benchmark datasets.

A. Datasets

We use nine datasets, including: 1) PIE [66]; 2) News4b [67]; 3) CBCL;[1] 4) Basehock;[2] 5) USPS49 [67]; 6) Lung;[2] 7) Leukemia;[2] 8) Gisette;[2] and 9) MNIST.[3] The details of the datasets are summarized in Table I.

B. Experimental Setup

We compare BSFS with the following methods.

1) *AllFea*, which uses all features for clustering.
2) *FSASL* [29], which adaptively learns the local and global structure when selecting features.
3) *SOGFS* [64], which learns an optimal graph structure for feature selection.
4) *LRPFS* [48], which aims to preserve the low-rank structure of data when selecting features.
5) *URAFS* [68], which uses generalized uncorrelated regression to select features and adaptively construct the graph.
6) *NSSLFS* [65], which selects features with sparse subspace learning.
7) *UGFS* [28], which is a top-k feature selection method considering combination effect.
8) *DGUFS* [26], which is a dependence guided feature selection method.
9) *RSOGFS* [36], which is a top-k feature selection method with optimal graph learning.
10) *FSBC* [58], which is a k-means-based balanced feature selection method.
11) *BSFS_nb*. To demonstrate the effectiveness of the balanced regularized term, we remove this term (or equivalently speaking, set $\gamma = 0$), leading to BSFS_nb.

To evaluate the quality of the selected features, we run k-means clustering on the selected features, and report the accuracy (ACC) and normalized mutual information (NMI) results. In addition, to evaluate the balance of the clustering results, we also report the normalized entropy (NE) [13], [14], which is defined as

$$\mathrm{NE} = -\frac{1}{\log(c)}\sum_{j=1}^{c}\frac{n_j}{n}\log\left(\frac{n_j}{n}\right) \tag{17}$$

where n is the number of instances, n_j is the number of instances in the jth cluster, and c is the number of clusters. From (17), we can find that $0 \le NE \le 1$, and the larger NE is, the more balanced the result is.

Since we often do not know the optimal number of selected features in advance, we report the results of different numbers of selected features [for the datasets except MNIST, in the range $\{10, 20, \ldots, 100\}$; and for MNIST, in the range $(\{10, 20, \ldots, 300\})$]. We also report the average results over the entire range of the number of selected features. In BSFS, we construct 10-nn graph $\tilde{\mathbf{S}}$ from the Gaussian kernel matrix $k(\mathbf{x}_i, \mathbf{x}_j) = e^{-[(\|\mathbf{x}_i - \mathbf{x}_j\|_2^2)/(2\sigma^2)]}$, where the bandwidth parameter σ is set as the average distance of all instance pairs. We tune γ in $[10^{-5}, 10^5]$ by grid search. For other compared methods, we tune the parameters as suggested in their papers.

All experiments are conducted using MATLAB 2017b on a PC computer with Windows 10, 3.41-GHz CPU and 32-GB memory.

C. Experimental Results on Clustering Performance

We report the average clustering results (ACC, NMI, and EN) over the range of selected features in Table II. Note that for the largest dataset MNIST, FSASL, SOGFS, LRPFS, URAFS, and DGUFS run out of memory. From Table II, we find that on most datasets, our method can outperform other unsupervised feature selection methods not only on the clustering performance (i.e., ACC and NMI) but also on the balance (i.e., EN). The main reasons may be in two-fold as follows.

1) First, we introduce the balanced regularized term to control the balance, which may improve the balance of the clustering results. Compared with BSFS_nb, which removes this term, our method performs better in most cases, and it demonstrates the effectiveness of this term.
2) Second, different from the compared methods which use $\ell_{2,1}$-norm to select features, our method uses $\ell_{2,0}$-norm to directly pick the exact top-k features, which makes the projection matrix $\mathbf{W}$ as sparsity as possible. The experimental results also demonstrate that this is more desirable for feature selection.

Figs. 1–3 show the detail results w.r.t. the different numbers of features. The horizontal yellow line represents the results of AllFea. From these figures, we can find that BSFS can outperform AllFea at most time. It means that although BSFS uses less features for clustering, it can still improve the clustering accuracy and balance. Moreover, BSFS also performs better

[1] http://cbcl.mit.edu/software-datasets/FaceData2.html
[2] http://featureselection.asu.edu/files/datasets/BASEHOCK.mat
[3] http://yann.lecun.com/exdb/mnist/

TABLE II
CLUSTERING RESULTS ON DIFFERENT DATASETS

Data	Metric	AllFea	FSASL	SOGFS	LRPFS	URAFS	NSSLFS	UGFS	DGUFS	RSOGFS	FSBC	BSFS_nb	BSFS
PIE	ACC	0.3197	0.4129	0.3761	0.4341	0.4740	0.2961	0.2871	0.2672	0.3672	0.3169	0.4622	**0.5196**
	NMI	0.6261	0.7067	0.6863	0.7310	0.7370	0.6008	0.6023	0.5384	0.6782	0.5899	0.7374	**0.7723**
	NE	0.9648	0.9665	0.9652	0.9676	0.9664	0.9676	**0.9714**	0.9512	0.9683	0.9673	0.9677	0.9709
News4b	ACC	0.2577	0.2691	0.2614	0.2545	0.2567	0.2579	0.2567	0.2550	0.2780	0.2568	0.2732	**0.2932**
	NMI	0.0062	0.0072	0.0074	0.0016	0.0056	0.0035	0.0048	0.0009	0.0235	0.0039	0.0174	**0.0369**
	NE	0.0597	0.1933	0.0313	0.0105	0.0471	0.1046	0.0316	0.0058	0.2166	0.0302	0.1938	**0.2485**
CBCL	ACC	0.6073	0.6861	0.6372	0.6737	0.6169	0.5948	0.5802	0.6715	0.5745	0.5237	0.5564	**0.7302**
	NMI	0.0416	0.1031	0.0709	0.0909	0.0456	0.0325	0.0269	0.0884	0.0279	0.0028	0.0100	**0.1612**
	NE	0.9509	0.9966	0.9692	0.9833	0.9809	0.9445	0.9274	0.9955	0.9464	0.9055	0.9636	**0.9982**
Basehock	ACC	0.5020	0.5904	0.5010	0.5171	0.5887	0.5019	0.5025	0.5008	0.5293	0.5020	0.5878	**0.5972**
	NMI	0.0026	0.0286	0.0021	0.0032	0.0286	0.0008	0.0018	0.0015	0.0068	0.0029	0.0284	**0.0326**
	NE	0.0251	0.8970	0.0217	0.3658	0.8808	0.0152	0.0362	0.0167	0.5696	0.0282	0.8813	**0.9138**
USPS49	ACC	0.6724	0.6034	0.6101	0.8011	0.6295	0.6863	0.6640	0.5542	0.6644	0.6907	0.7983	**0.8110**
	NMI	0.1060	0.0703	0.0543	**0.3619**	0.0563	0.2541	0.0964	0.0336	0.1231	0.1274	0.2758	0.3066
	NE	0.9105	0.6537	0.7803	0.9270	0.9392	0.7548	0.9178	0.4306	0.8952	0.9339	**0.9996**	0.9922
Lung	ACC	0.6274	0.6189	0.6295	0.5711	0.6223	0.5895	0.5989	0.5892	0.5859	0.6329	0.6534	**0.6704**
	NMI	0.3789	0.6046	0.5914	0.5365	0.5890	0.5512	0.5565	0.5341	0.5507	0.6035	0.6224	**0.6390**
	NE	0.8736	0.9260	0.9126	0.9288	0.9352	0.9214	0.9268	0.9126	0.9152	0.9346	0.9335	**0.9393**
Leukemia	ACC	0.6583	0.6961	0.5769	0.6587	0.5988	0.5624	0.6200	0.6501	0.6537	0.6611	0.6781	**0.6971**
	NMI	0.0858	0.1115	0.0243	0.0669	0.0824	0.0181	0.0520	0.0561	0.0778	0.1028	0.0961	**0.1126**
	NE	0.9948	0.9898	0.9883	0.9790	0.9391	0.9859	0.9898	0.3307	0.9872	0.9654	0.9969	**0.9982**
Gisette	ACC	0.6794	0.7273	0.5427	0.6227	0.5787	0.5331	0.5659	0.5448	0.5356	0.6074	0.8223	**0.8303**
	NMI	0.1043	0.1879	0.0184	0.1068	0.0265	0.0094	0.0438	0.0162	0.0173	0.0731	0.3511	**0.3679**
	NE	0.9422	0.9063	0.4254	0.6808	0.8318	0.6395	0.5286	0.6832	0.6304	0.7916	0.9671	**0.9708**
MNIST	ACC	0.5425	-	-	-	-	0.4026	-	-	-	0.4592	0.4638	**0.4662**
	NMI	0.5245	-	-	-	-	0.3221	-	-	-	0.4155	0.4218	**0.4230**
	NE	0.9761	-	-	-	-	0.9593	-	-	-	**0.9860**	0.9800	0.9824

TABLE III
RUNNING TIME OF ALL METHODS ON DIFFERENT DATASETS TO SELECT 100 FEATURES (SEC.)

Data sets	FSASL	SOGFS	LRPFS	URAFS	NSSLFS	UGFS	DGUFS	RSOGFS	FSBC	BSFS
PIE	11.882 (26×)	25.453 (12×)	91.893 (3.4×)	22.255 (14×)	313.81 (1×)	20.625 (15.2×)	111.95 (2.8×)	11.256 (27.9×)	15.347 (20×)	**6.3941** **(49×)**
News4b	175.34 (64×)	5281.4 (2.1×)	1192.6 (9.4×)	432.74 (26×)	11225 (1×)	56.562 (198.5×)	2757.5 (4.1×)	1312.7 (8.6×)	630.52 (18×)	**44.581** **(252×)**
CBCL	16.104 (16×)	20.226 (13×)	256.12 (1×)	10.884 (24×)	91.791 (2.8×)	6.0946 (42×)	212.92 (1.2×)	**2.9498** **(72.2×)**	5.2805 (49×)	3.4607 (74×)
Basehock	77.229 (67×)	3596.6 (1.4×)	353.86 (15×)	251.65 (20×)	5190.1 (1×)	13.234 (392×)	516.60 (10×)	724.99 (7.2×)	404.59 (13×)	**9.7885** **(530×)**
USPS49	8.9413 (16×)	8.5662 (17×)	144.89 (1×)	8.0380 (18×)	53.978 (2.7×)	4.3014 (33.7×)	123.42 (1.2×)	3.0332 (47.8×)	3.7384 (39×)	**2.3650** **(61×)**
Lung	0.4005 (21×)	2.2669 (3.8×)	0.1730 (50×)	0.3839 (22×)	8.5988 (1×)	**0.0317** **(271×)**	0.1927 (45×)	0.4340 (19.8×)	0.8402 (10×)	0.3551 (24×)
Leukemia	133.73 (205×)	27507 (1×)	14.443 (1905×)	444.08 (62×)	2654.5 (10×)	9.1195 (3016×)	3.4856 (7891×)	2351.0 (11.7×)	957.38 (28×)	**0.9503** **(28946×)**
Gisette	456.25 (31×)	2220.5 (6.4×)	4085.7 (3.5×)	485.47 (29×)	14237 (1×)	97.701 (145.7×)	10843 (1.3×)	1454.9 (9.8×)	507.90 (28×)	**84.694** **(168×)**
MNIST	- -	- -	- -	- -	11143 (1×)	- -	- -	- -	6287.9 (1.8×)	**937.37** **(11.9×)**

than other state-of-the-art algorithms in most cases, which well demonstrates its superiority.

Fig. 4 shows the selected features of our method in PIE dataset, which is a face dataset. We show the results of $20, 40, \ldots, 100$ selected features. The yellow points are selected features. From Fig. 4, we find that our method prefers to select the features in some discriminative parts, such as eyes, nose, and mouth, which can well describe the character of a person.

Fig. 5 shows an example result of the numbers of instances in each cluster on PIE dataset. Fig. 5(a) shows the result of kmeans on all features. Fig. 5(b)–(f) shows the results with $20, 40, \ldots, 100$ selected features. We can see that with the selected features, the data are more balanced than the data with all features, which demonstrates the effectiveness of our balance schema.

D. Experimental Results on Efficiency

To demonstrate the efficiency of the proposed method, we show the running time of all methods to select 100 features in Table III. From Table III, we can find that our method is significantly faster than other compared methods, especially on some high-dimensional data. For example, we achieve 28 946 times and 530 times speedup compared with the slowest methods on Leukemia and Basehock datasets, respectively. Due to

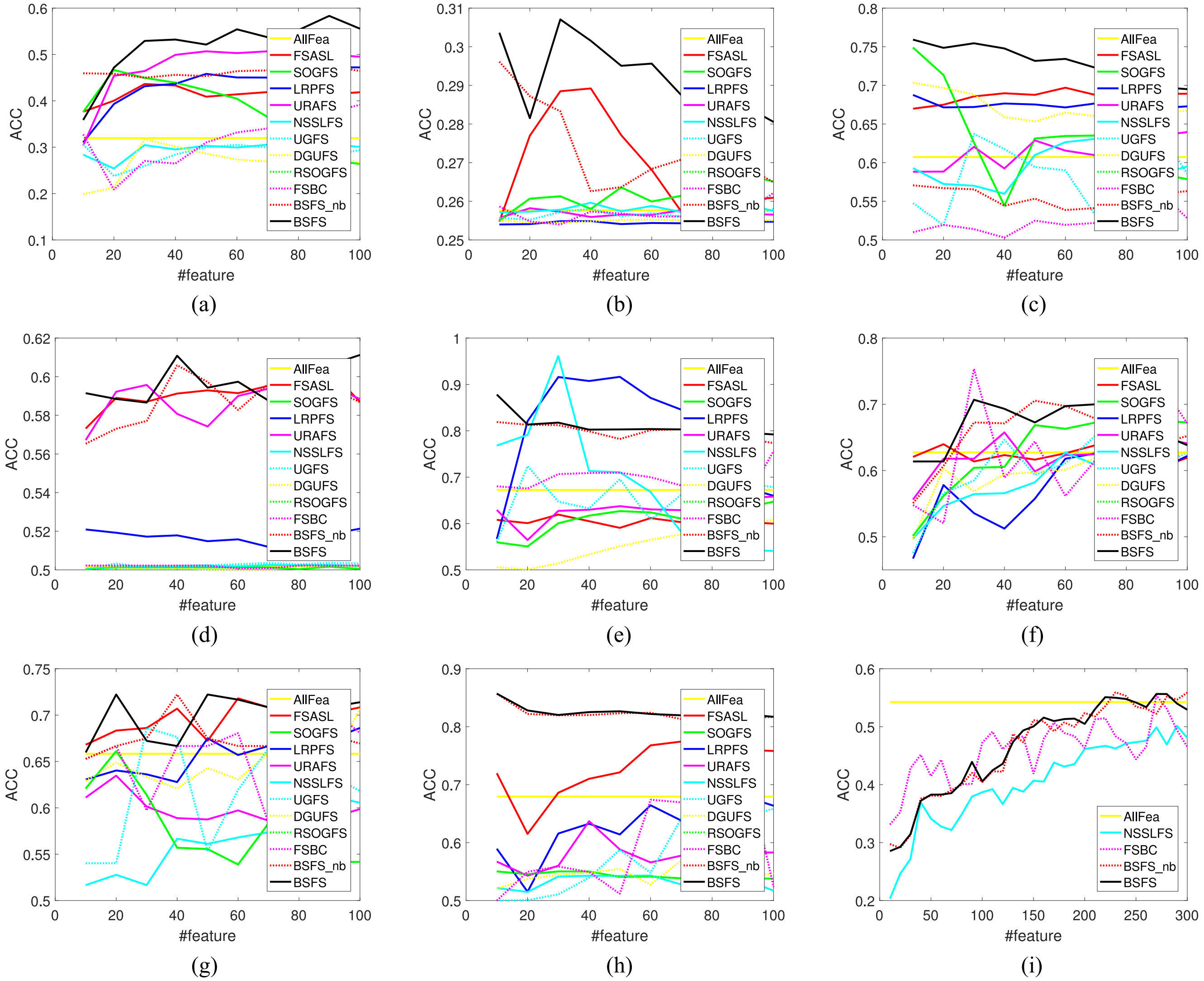


Fig. 1. ACC results on all datasets with different number of features. (a) ACC on PIE. (b) ACC on News4b. (c) ACC on CBCL. (d) ACC on Basehock. (e) ACC on USPS. (f) ACC on Lung. (g) ACC on Leukemia. (h) ACC on Gisette. (i) ACC on MNIST.

the proposed speedup strategy, we reduce the time complexity from cubic with d to linear with d, which makes our method faster than the compared ones.

Fig. 6 shows the convergence curves of BSFS on PIE, News4b, CBCL, and Lung datasets. The results on other datasets are similar. From Fig. 6, we find that BSFS often converges within several tens iterations.

E. Experiments on Balanced Clustering

As one application, our method can be used for balance clustering. To demonstrate the effectiveness, we compare our BSFS (with 300 selected features for MNIST and 100 selected features for other datasets) with some balanced clustering methods in this section. The compared balanced clustering methods are as follows.

1) *ISC* [69], which applies the size constraint to guarantee the balance. It is a hard-balanced clustering.
2) *BCLS* [13], which is a balanced clustering with least square regression. It is a soft-balanced clustering.
3) *BCKM* [14], which is a balanced clustering with kmeans. It is a soft-balanced clustering.
4) *BCMC* [14], which is a balanced clustering with min-cut. It is a soft-balanced clustering.

Table IV shows the ACC, NMI, and NE results on all datasets. Note that on the largest dataset MNIST, ISC and BCLS run out of memory. On most datasets, our method can achieve better clustering performance on ACC and NMI. It demonstrates the discriminant of the selected features of our method. With respect to the balance metric NE, ISC often achieves the best performance, because it is a hard-balanced clustering method. Our method is not better than the soft-balanced methods, but it is closed to them on most datasets. The reason may be that in the balanced clustering methods, they often impose a balanced constraint directly on the clustering results. Notice that our method is a feature selection method, that is, we only select some features and then in the clustering process, we only perform the regular kmeans on the selected features without any balanced constraints. Therefore, the balanced constraint is implicit in our method and is not

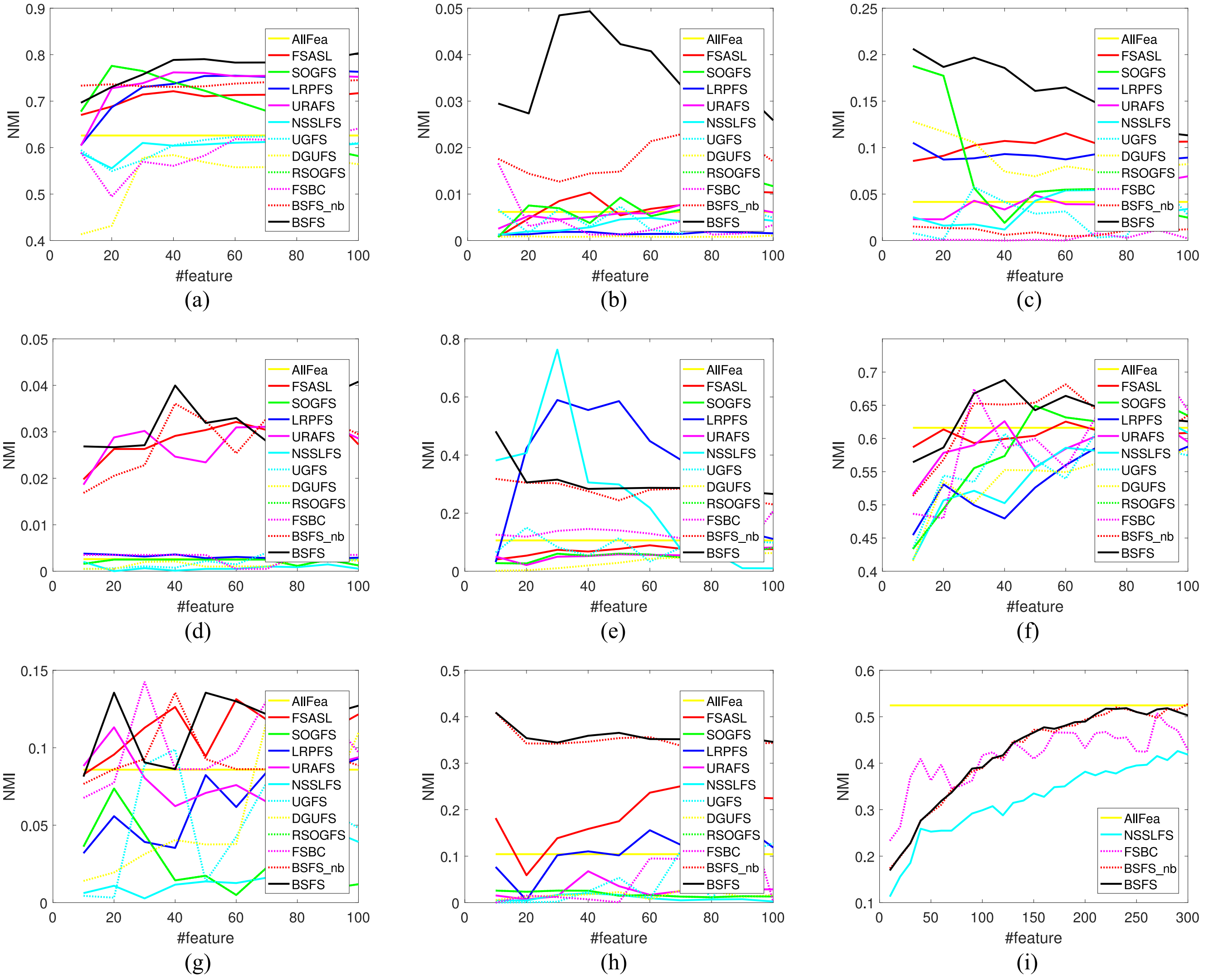
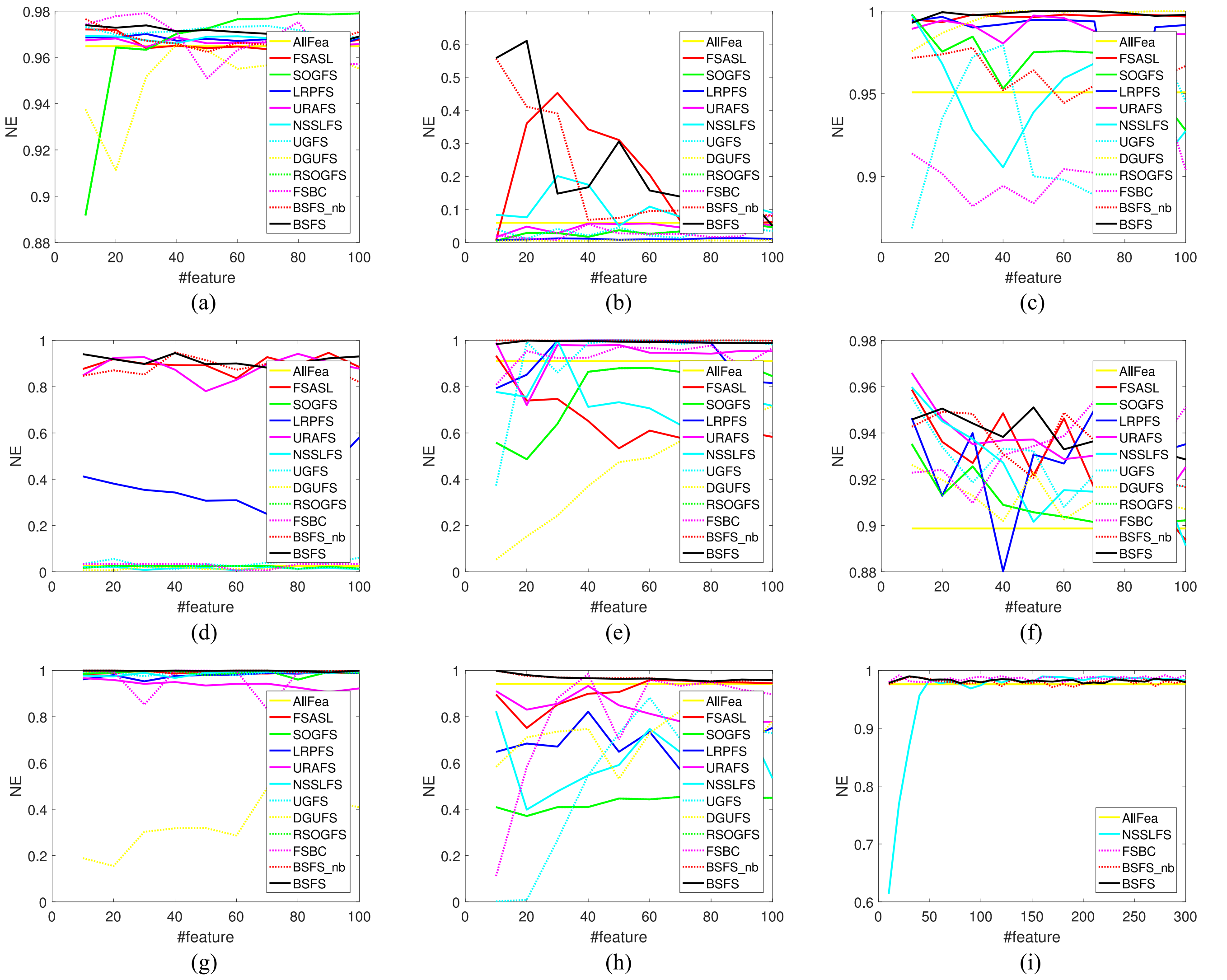


Fig. 2. NMI results on all datasets with different number of features. (a) NMI on PIE. (b) NMI on News4b. (c) NMI on CBCL. (d) NMI on Basehock. (e) NMI on USPS. (f) NMI on Lung. (g) NMI on Leukemia. (h) NMI on Gisette. (i) NMI on MNIST.

TABLE IV
CLUSTERING RESULTS COMPARED WITH BALANCED CLUSTERING METHODS

Methods	metric	PIE	News4b	CBCL	Basehock	USPS49	Lung	Leukemia	Gisette	MNIST
All features+ISC	ACC	0.3193	**0.4584**	0.5540	0.5665	0.7591	0.4932	0.6528	0.6403	-
	NMI	0.5584	**0.1253**	0.0084	0.0128	0.2036	0.3950	0.0762	0.0576	-
	NE	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	-
All features+BCLS	ACC	0.1015	0.2682	0.6275	0.5143	0.5146	0.2740	0.5833	0.5001	-
	NMI	0.3256	0.0011	0.0474	0.0000	0.0001	0.1771	0.0097	0.0000	-
	NE	0.9986	0.9995	**1**	**1**	**1**	0.9702	0.9726	**1**	-
All features+BCKM	ACC	0.3564	0.3343	0.5360	0.6116	0.7561	0.6164	0.6528	0.5111	0.4871
	NMI	0.6734	0.0302	0.0037	0.0363	0.1986	0.6106	0.0762	0.0001	0.4271
	NE	0.9975	**1**	**1**	**1**	**1**	0.9872	**1**	**1**	**1**
All features+BCMC	ACC	0.3151	0.2956	0.5160	0.5855	0.5093	**0.8356**	0.6250	0.5000	0.1089
	NMI	0.6546	0.0168	0.0001	0.0212	0.0000	**0.7579**	0.0380	0.0000	0.0211
	NE	0.9461	0.3760	**1**	0.9994	0.0133	0.9441	0.9911	0.0000	0.0027
BSFS+kmeans	ACC	**0.5675**	0.2809	**0.6952**	**0.6126**	**0.7923**	0.7027	**0.7153**	**0.8237**	**0.5292**
	NMI	**0.8040**	0.0339	**0.1134**	**0.0408**	**0.2667**	0.6597	**0.1285**	**0.3578**	**0.5030**
	NE	0.9698	0.1577	0.9977	0.9450	0.9880	0.9400	0.9993	0.9638	0.9819

as strong as those balanced clustering. Despite this, the NE of our method is closed to the balanced clustering methods on most datasets, which demonstrates that the selected features can indeed reveal the balanced structure. Notice that different from the balanced clustering methods, which are designed for particular clustering methods, our method provides a more



(a) (b) (c) (d) (e) (f) (g) (h) (i)

Fig. 3. NE results on all datasets with different number of features. (a) NE on PIE. (b) NE on News4b. (c) NE on CBCL. (d) NE on Basehock. (e) NE on USPS. (f) NE on Lung. (g) NE on Leukemia. (h) NE on Gisette. (i) NE on MNIST.

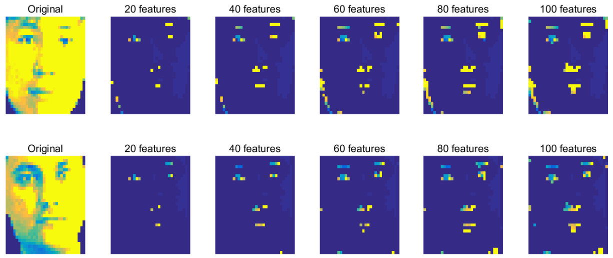


Fig. 4. Example result of selected features (with different numbers of features) of BSFS on PIE dataset.

general way to reveal the balanced structure and can be followed by any standard data analyzing method such as *k*-means. Moreover, the experimental results may reveal that sometimes the explicit balanced constraints used in existing balanced clustering methods may affect the clustering performance, and the early processing method like ours may achieve a better tradeoff between the clustering performance and balance. Therefore, the balanced feature selection may be more appropriate for discovering the balanced structure sometimes.

F. Parameter Study

In our method, there is only one hyperparameter (γ), which is needed to tune manually. To explore the effect of γ on clustering performance and balance, we tune it in the range $[10^{-5}, 10^{5}]$. We show the ACC, NMI, and EN results on PIE and CBCL datasets in Fig. 7. The results on other datasets are similar. From Fig. 7, we can see that our BSFS can achieve a relatively stable performance in a wide range of γ. Therefore, the selection of the parameter γ is not difficult in practice.

V. Conclusion

In this article, we proposed a novel BSFS method. Different from conventional unsupervised feature selection methods, which focused on selecting the discriminative features, our method also paid attention to the balanced structure of data. By introducing the balanced regularized term, we integrated

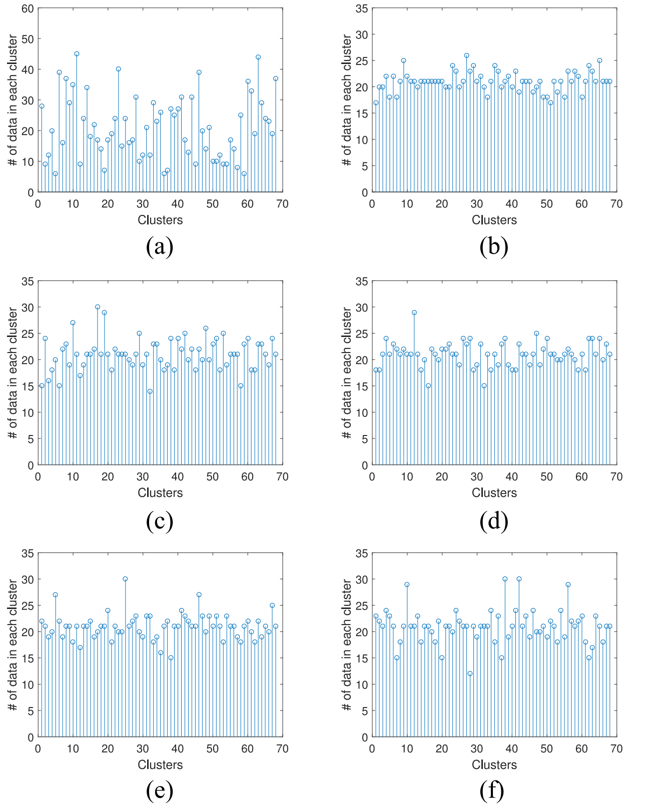


Fig. 5. Example result of the numbers of data in each cluster (with different numbers of features) of BSFS on PIE dataset. (a) Original data. (b) 20 selected features. (c) 40 selected features. (d) 60 selected features. (e) 80 selected features. (f) 100 selected features.

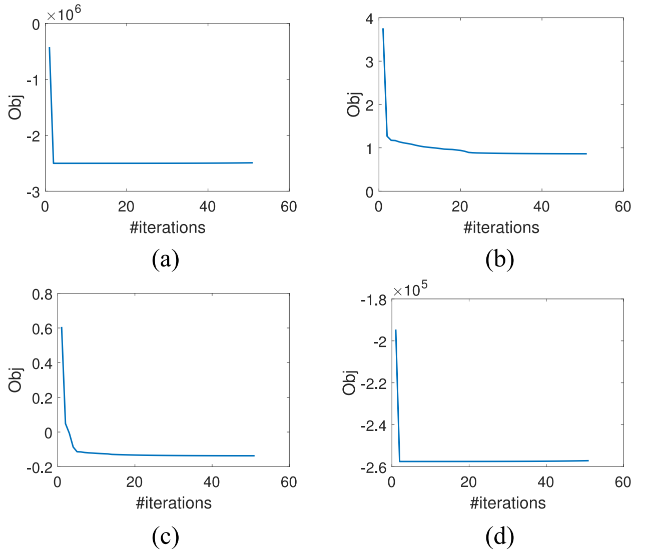


Fig. 6. Convergence curves on (a) PIE, (b) News4b, (c) CBCL, and (d) Lung.

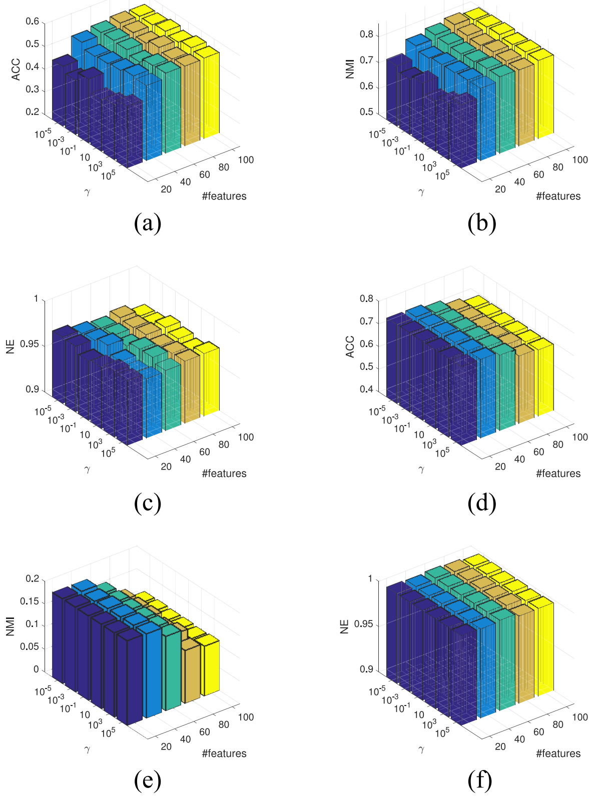


Fig. 7. Clustering results on PIE and CBCL with respect to different values of γ. (a) ACC on PIE. (b) NMI on PIE. (c) EN on PIE. (d) ACC on CBCL. (e) NMI on CBCL. (f) EN on CBCL.

the balanced spectral clustering and feature selection method into a unified framework. Then, we proposed an effective and efficient ADMM method to optimize the objective function, which can select the exact top-k features without any postprocessing. The extensive experimental results show that BSFS outperformed the state-of-the-art methods not only on the clustering performance but also on the balance of the clustering results.

The strengths of the proposed method are in two-fold. On the one hand, the proposed one is highly efficient compared with other state-of-the-art unsupervised feature selection method. On the other hand, on the balanced data, since the proposed method considers balance explicitly, it can better reveal the balanced structure of data. Despite these merits, notice that we make the assume that the given data is balanced. If the data itself is unbalanced, our method may be misled by the balance and achieve some worse results. Therefore, the proposed method may be inappropriate to handle the imbalanced data. In the future, we will study how to evaluate the balance of data without any label information, that is, given a data, we judge whether the proposed method will work before selecting features.

References

[1] Z. Du, Y. Liu, and D. Qian, "An energy-efficient balanced clustering algorithm for wireless sensor networks," in *Proc. 5th Int. Conf. Wireless Commun. Netw. Mobile Comput.*, 2009, pp. 1–4.

[2] W. Su, J. Hu, C. Lin, and S. Shen, "SLA-aware tenant placement and dynamic resource provision in SaaS," in *Proc. IEEE Int. Conf. Web Serv.*, 2015, pp. 615–622.

[3] F. Nie, H. Huang, X. Cai, and C. H. Ding, "Efficient and robust feature selection via joint $\ell_{2,1}$-norms minimization," in *Advances in Neural Information Processing Systems*. Red Hook, NY, USA: Curran, 2010, pp. 1813–1821.

[4] Z. Lai, D. Mo, W. K. Wong, Y. Xu, D. Miao, and D. Zhang, "Robust discriminant regression for feature extraction," *IEEE Trans. Cybern.*, vol. 48, no. 8, pp. 2472–2484, Aug. 2018.

[5] P. Zhu, W. Zuo, L. Zhang, Q. Hu, and S. C. K. Shiu, "Unsupervised feature selection by regularized self-representation," *Pattern Recognit.*, vol. 48, no. 2, pp. 438–446, 2015.

[6] L. Du, C. Ren, X. Lv, Y. Chen, P. Zhou, and Z. Hu, "Local graph reconstruction for parameter free unsupervised feature selection," *IEEE Access*, vol. 7, pp. 102921–102930, 2019.

[7] D. Ming and C. Ding, "Robust flexible feature selection via exclusive L21 regularization," in *Proc. 28th Int. Joint Conf. Artif. Intell. (IJCAI)*, Jul. 2019, pp. 3158–3164.

[8] G. Roffo, S. Melzi, U. Castellani, A. Vinciarelli, and M. Cristani, "Infinite feature selection: A graph-based feature filtering approach," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 12, pp. 4396–4410, Dec. 2021.

[9] C. Tang *et al.*, "Unsupervised feature selection via latent representation learning and manifold regularization," *Neural Netw.*, vol. 117, pp. 163–178, Sep. 2019.

[10] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Advances in Neural Information Processing Systems*. Cambridge, MA, USA: MIT Press, 2002, pp. 849–856.

[11] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends® Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.

[12] S. Zhong and J. Ghosh, "Model-based clustering with soft balancing," in *Proc. ICDM*, 2003, pp. 459–466.

[13] H. Liu, J. Han, F. Nie, and X. Li, "Balanced clustering with least square regression," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 2231–2237.

[14] Z. Li, F. Nie, X. Chang, Z. Ma, and Y. Yang, "Balanced clustering via exclusive lasso: A pragmatic approach," in *Proc. 32nd AAAI Conf. Artif. Intell/*, 2018, pp. 3596–3603.

[15] M. Fan, X. Chang, X. Zhang, D. Wang, and L. Du, "Top-k supervise feature selection via ADMM for integer programming," in *Proc. IJCAI*, 2017, pp. 1646–1653.

[16] S. Liao, Q. Gao, F. Nie, Y. Liu, and X. Zhang, "Worst-case discriminative feature selection," in *Proc. IJCAI*, 2019, pp. 2973–2979.

[17] J. G. Dy and C. E. Brodley, "Feature selection for unsupervised learning," *J. Mach. Learn. Res.*, vol. 5, pp. 845–889, Dec. 2004.

[18] C. Feng, C. Qian, and K. Tang, "Unsupervised feature selection by Pareto optimization," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, Jul. 2019, pp. 3534–3541.

[19] L. Teng *et al.*, "Unsupervised feature selection with adaptive residual preserving," *Neurocomputing*, vol. 367, pp. 259–272, Nov. 2019.

[20] Y. Han, Y. Yang, Y. Yan, Z. Ma, N. Sebe, and X. Zhou, "Semisupervised feature selection via spline regression for video semantic recognition," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 2, pp. 252–264, Feb. 2015.

[21] X. Chang and Y. Yang, "Semisupervised feature analysis by mining correlations among multiple tasks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2294–2305, Oct. 2016.

[22] P. Zhu, Q. Xu, Q. Hu, and C. Zhang, "Co-regularized unsupervised feature selection," *Neurocomputing*, vol. 275, pp. 2855–2863, Jan. 2018.

[23] S. Feng and M. F. Duarte, "Graph autoencoder-based unsupervised feature selection with broad and local data structure preservation," *Neurocomputing*, vol. 312, pp. 310–323, Oct. 2018.

[24] T. Xie, P. Ren, T. Zhang, and Y. Y. Tang, "Distribution preserving learning for unsupervised feature selection," *Neurocomputing*, vol. 289, pp. 231–240, May 2018.

[25] X. Ye, H. Li, A. Imakura, and T. Sakurai, "Distributed collaborative feature selection based on intermediate representation," in *Proc. 28th Int. Joint Conf. Artif. Intell. (IJCAI)*, Macao, China, Aug. 2019, pp. 4142–4149.

[26] J. Guo and W. Zhu, "Dependence guided unsupervised feature selection," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 2232–2239.

[27] H. T. Shen, Y. Zhu, W. Zheng, and X. Zhu, "Half-quadratic minimization for unsupervised feature selection on incomplete data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 7, pp. 3122–3135, Jul. 2021.

[28] X. Du, F. Nie, W. Wang, Y. Yang, and X. Zhou, "Exploiting combination effect for unsupervised feature selection by $l_{2,0}$ norm," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 1, pp. 201–214, Jan. 2019.

[29] L. Du and Y.-D. Shen, "Unsupervised feature selection with adaptive structure learning," in *Proc. SIGKDD*, 2015, pp. 209–218.

[30] R. Shang, W. Wang, R. Stolkin, and L. Jiao, "Non-negative spectral learning and sparse regression-based dual-graph regularized feature selection," *IEEE Trans. Cybern.*, vol. 48, no. 2, pp. 793–806, Feb. 2018.

[31] R. Shang, K. Xu, F. Shang, and L. Jiao, "Sparse and low-redundant subspace learning-based dual-graph regularized robust feature selection," *Knowl. Based Syst.*, vol. 187, Jan. 2020, Art. no. 104830.

[32] X. Zhu, S. Zhang, R. Hu, Y. Zhu, and J. Song, "Local and global structure preservation for robust unsupervised spectral feature selection," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 3, pp. 517–529, Mar. 2018.

[33] X. Dong, L. Zhu, X. Song, J. Li, and Z. Cheng, "Adaptive collaborative similarity learning for unsupervised multi-view feature selection," in *Proc. 27th Int. Joint Conf. Artif. Intell. (IJCAI)*, Stockholm, Sweden, Jul. 2018, pp. 2064–2070.

[34] C. Tang, X. Zhu, J. Chen, P. Wang, X. Liu, and J. Tian, "Robust graph regularized unsupervised feature selection," *Expert Syst. Appl.*, vol. 96, pp. 64–76, Apr. 2018.

[35] P. Zhou, L. Du, X. Li, Y.-D. Shen, and Y. Qian, "Unsupervised feature selection with adaptive multiple graph learning," *Pattern Recognit.*, vol. 105, Sep. 2020, Art. no. 107375.

[36] F. Nie, X. Dong, L. Tian, R. Wang, and X. Li, "Unsupervised feature selection with constrained $l2,0$-norm and optimized graph," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Dec. 25, 2020, doi: 10.1109/TNNLS.2020.3043362.

[37] C. Tang *et al.*, "Robust unsupervised feature selection via dual self-representation and manifold regularization," *Knowl. Based Syst.*, vol. 145, pp. 109–120, Apr. 2018.

[38] C. Tang *et al.*, "Consensus learning guided multi-view unsupervised feature selection," *Knowl. Based Syst.*, vol. 160, pp. 49–60, Nov. 2018.

[39] Z. Zhang and E. R. Hancock, "Hypergraph based information-theoretic feature selection," *Pattern Recognit. Lett.*, vol. 33, no. 15, pp. 1991–1999, 2012.

[40] Z. Zhang, L. Bai, Y. Liang, and E. Hancock, "Joint hypergraph learning and sparse regression for feature selection," *Pattern Recognit.*, vol. 63, pp. 291–309, Mar. 2017.

[41] F. Luo, B. Du, L. Zhang, L. Zhang, and D. Tao, "Feature learning using spatial-spectral hypergraph discriminant analysis for hyperspectral image," *IEEE Trans. Cybern.*, vol. 49, no. 7, pp. 2406–2419, Jul. 2019.

[42] F. Luo, L. Zhang, B. Du, and L. Zhang, "Dimensionality reduction with enhanced hybrid-graph discriminant learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 8, pp. 5336–5353, Aug. 2020.

[43] Y. Duan, H. Huang, and Y. Tang, "Local constraint-based sparse manifold hypergraph learning for dimensionality reduction of hyperspectral image," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 1, pp. 613–628, Jan. 2021.

[44] J. Wen, Z. Lai, Y. Zhan, and J. Cui, "The $L_{2,1}$-norm-based unsupervised optimal feature selection with applications to action recognition," *Pattern Recognit.*, vol. 60, pp. 515–530, Dec. 2016.

[45] P. Zhu, W. Zhu, Q. Hu, C. Zhang, and W. Zuo, "Subspace clustering guided unsupervised feature selection," *Pattern Recognit.*, vol. 66, pp. 364–374, Jun. 2017.

[46] M. Fan, X. Chang, and D. Tao, "Structure regularized unsupervised discriminant feature analysis," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 1870–1876.

[47] X. Zhang, M. Fan, D. Wang, P. Zhou, and D. Tao, "Top-k feature selection framework using robust 0–1 integer programming," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 7, pp. 3005–3019, Jul. 2021.

[48] W. Zheng, C. Xu, J. Yang, J. Gao, and F. Zhu, "Low-rank structure preserving for unsupervised feature selection," *Neurocomputing*, vol. 314, pp. 360–370, Nov. 2018.

[49] C. Tang *et al.*, "Feature selective projection with low-rank embedding and dual Laplacian regularization," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 9, pp. 1747–1760, Sep. 2020.

[50] S. Li, C. Tang, X. Liu, Y. Liu, and J. Chen, "Dual graph regularized compact feature representation for unsupervised feature selection," *Neurocomputing*, vol. 331, pp. 77–96, Feb. 2019.

[51] G. Zhong and C.-M. Pun, "Subspace clustering by simultaneously feature selection and similarity learning," *Knowl. Based Syst.*, vol. 193, Apr. 2020, Art. no. 105512.

[52] L. Du, Z. Shen, X. Li, P. Zhou, and Y.-D. Shen, "Local and global discriminative learning for unsupervised feature selection," in *Proc. IEEE 13th Int. Conf. Data Min.*, 2013, pp. 131–140.

[53] C. Hou, F. Nie, X. Li, D. Yi, and Y. Wu, "Joint embedding learning and sparse regression: A framework for unsupervised feature selection," *IEEE Trans. Cybern.*, vol. 44, no. 6, pp. 793–804, Jun. 2014.

[54] S. Wang, J. Tang, and H. Liu, "Embedded unsupervised feature selection," in *Proc. 29th AAAI Conf. Artif. Intell.*, 2015, pp. 470–476.

[55] H. Zhang, R. Zhang, F. Nie, and X. Li, "An efficient framework for unsupervised feature selection," *Neurocomputing*, vol. 366, pp. 194–207, Nov. 2019.

[56] J. Chen, Y. Zeng, Y. Li, and G.-B. Huang, "Unsupervised feature selection based extreme learning machine for clustering," *Neurocomputing*, vol. 386, pp. 208–220, Apr. 2020.

[57] C. Tang *et al.*, "Cross-view locality preserved diversity and consensus learning for multi-view unsupervised feature selection," *IEEE Trans. Knowl. Data Eng.*, early access, Jan. 1, 2021, doi: 10.1109/TKDE.2020.3048678.

[58] P. Zhou, J. Chen, M. Fan, L. Du, Y.-D. Shen, and X. Li, "Unsupervised feature selection for balanced clustering," *Knowl. Based Syst.*, vol. 193, Apr. 2020, Art. no. 105417.

[59] P. Bradley, K. Bennett, and A. Demiriz, *Constrained K-Means Clustering*, vol. 20, Microsoft Res., Redmond, WA, USA, 2000.

[60] M. I. Malinen and P. Fränti, "Balanced *K*-means for clustering," in *Proc. Joint IAPR Int. Workshops Stat. Techn. Pattern Recognit. (SPR) Struct. Syntactic Pattern Recognit. (SSPR)*, 2014, pp. 32–41.

[61] L. R. Costa, D. Aloise, and N. Mladenović, "Less is more: Basic variable neighborhood search heuristic for balanced minimum sum-of-squares clustering," *Inf. Sci.*, vols. 415–416, pp. 247–253, Nov. 2017.

[62] A. Banerjee and J. Ghosh, "On scaling up balanced clustering algorithms," in *Proc. SIAM Int. Conf. Data Min.*, 2002, pp. 333–349.

[63] A. Banerjee and J. Ghosh, "Frequency-sensitive competitive learning for scalable balanced clustering on high-dimensional hyperspheres," *IEEE Trans. Neural Netw.*, vol. 15, no. 3, pp. 702–719, May 2004.

[64] F. Nie, W. Zhu, and X. Li, "Unsupervised feature selection with structured graph optimization," in *Proc. AAAI*, 2016, pp. 1302–1308.

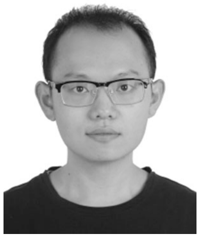
[65] W. Zheng, H. Yan, and J. Yang, "Robust unsupervised feature selection by nonnegative sparse subspace learning," *Neurocomputing*, vol. 334, pp. 156–171, Mar. 2019.

[66] T. Sim, S. Baker, and M. Bsat, "The CMU pose, illumination, and expression (PIE) database," in *Proc. 5th IEEE Int. Conf. Autom. Face Gesture Recognit.*, 2002, pp. 53–58.

[67] M. Wu and B. Schölkopf, "A local learning approach for clustering," in *Advances in Neural Information Processing Systems*. Cambridge, MA, USA: MIT Press, 2007, pp. 1529–1536.

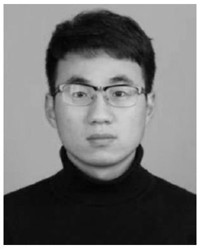
[68] X. Li, H. Zhang, R. Zhang, Y. Liu, and F. Nie, "Generalized uncorrelated regression with adaptive graph for unsupervised feature selection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 5, pp. 1587–1595, May 2019.

[69] S. Zhu, D. Wang, and T. Li, "Data clustering with size constraints," *Knowl. Based Syst.*, vol. 23, no. 8, pp. 883–889, 2010.



Peng Zhou (Member, IEEE) received the B.E. degree in computer science and technology from the University of Science and Technology of China, Hefei, China, in 2011, and the Ph.D degree in computer science from the Institute of Software, Chinese Academy of Sciences, Beijing, China, in 2017.

He is currently an Associate Professor with the School of Computer Science and Technology, Anhui University, Hefei. He has published more than 20 papers in highly regarded conferences and journals, including IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, *Pattern Recognition*, *Information Fusion*, IJCAI, AAAI, SDM, and ICDM. More publications and codes can be found in his homepage: https://doctor-nobody.github.io/. His research interests include machine learning, data mining, and artificial intelligence.



Jiangyong Chen is currently pursuing the M.S. degree with the School of Computer Science and Technology, Anhui University, Hefei, China.

His research interests include machine learning and data mining.

Liang Du (Member, IEEE) received the B.E. degree in software engineering from Wuhan University, Wuhan, China, in 2007, and the Ph.D. degree in computer science from the Institute of Software, University of Chinese Academy of Sciences, Beijing, China, in 2013.

From July 2013 to July 2014, he was a Software Engineer with Alibaba Group, Hangzhou, China. He was also an Assistant Researcher with the State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing. He is currently a Lecturer with Shanxi University, Taiyuan, China. He has published more than 40 papers in top conferences and journals, including KDD, IJCAI, AAAI, ICDM, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, SDM, and CIKM. His research interests include clustering with noise and heterogeneous data, ranking for feature selection, active learning, and document summarization.

Xuejun Li (Member, IEEE) received the Ph.D. degree from Anhui University, Hefei, China, in 2008.

He is currently a Professor with the School of Computer Science and Technology, Anhui University. His major research interests include workflow systems, cloud computing, and intelligent software.