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
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 \( \ell_1 \) 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 $A \in \mathbb{R}^{m \times n}$, $A_{ij}$ denotes its $(i, j)$th element.

A. Formulation

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

$$
\min_{W} \|Y - XW\|_F^2
$$

s.t. $\|W\|_{2,0} = k$

(1)

where $\|W\|_{2,0}$ is the $\ell_{2,0}$-norm, which counts the number of nonzero rows in $W$. Since we aim to select $k$ features, we impose the $\ell_{2,0}$-norm constraint on $W$ to make sure that there are exact $k$ nonzero rows in $W$. 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 $Y$ in advance, which makes the problem more challenging. A natural idea is that we can use some clustering methods to generate the pseudolabels $Y$. 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 $S \in \mathbb{R}^{n \times n}$, where $S_{ij} \geq 0$ is the similarity of the $i$th and the $j$th instances. Then, we obtain the normalized Laplacian matrix $L \in \mathbb{R}^{n \times n}$ by $L = I - D^{-1/2}SD^{-1/2}$, where $I$ is an identity matrix and $D \in \mathbb{R}^{n \times n}$ is a diagonal matrix whose diagonal elements are $D_{ii} = \sum_{j=1}^{n} S_{ij}$. Spectral clustering aims to learn an orthogonal embedding $Y \in \mathbb{R}^{n \times c}$ of $L$ by optimizing

$$
\min_{Y} \text{tr}(Y^TLY)
$$

s.t. $Y^TY = I$.

(2)

Note that in the traditional spectral clustering, it relaxes the label matrix into a real-valued orthogonal matrix $Y$. 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:

$$
\min_{Y} \text{tr}(Y^TLY)
$$

s.t. $Y \in \{0, 1\}^{n \times c}$, $\sum_{j=1}^{c} Y_{ij} = 1$

(3)
where \(\sum_{j=1}^{c} Y_j = 1\) makes sure that each instance just belongs to one cluster. Taking the definition of \(L\) into (3), we have
\[
\text{tr}(Y^TLY) = \text{tr}(Y^TD^{-1/2}SD^{-1/2}Y) = n - \text{tr}(Y^TD^{-1/2}SD^{-1/2}Y).
\]

Let \(\tilde{S} = D^{-1/2}SD^{-1/2}\), from (4), we have \(\min \text{tr}(Y^TLY)\) is equivalent to \(\max \text{tr}(Y^T\tilde{S}Y)\). Combining (1) and (3), we obtain the following multiobjective optimization problem:
\[
\begin{aligned}
& \min W \quad ||Y - XW||_F^2, \\
& \max Y \quad \text{tr}(Y^T\tilde{S}Y).
\end{aligned}
\]

To optimize this multiobjective optimization problem without introducing any more hyperparameters, we rewrite it as the following form:
\[
\begin{aligned}
& \min_{Y,W} \frac{||Y - XW||_F^2}{\text{tr}(Y^T\tilde{S}Y)} \\
& \text{s.t.} \quad ||W||_2,0 = k, \quad Y \in \{0, 1\}^{n \times c}, \quad \sum_{j=1}^{c} Y_{ij} = 1.
\end{aligned}
\]

Note that \(Y\) plays the role as the pseudolabels, so we can achieve a balanced clustering result by imposing some constraints on \(Y\). 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 \(Y\) by columns, we can obtain the number of instances in every cluster. We denote the number of instances in the \(j\)th 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 \(p \in \mathbb{R}^c\) of the number of instances in each cluster by \(p_j = (n_j/n)\). Since we wish the distribution \(p\) should be as balanced as possible, we can achieve this by maximizing the Shannon entropy of the distribution \(p\), that is, \(\max p - \sum_{j=1}^{c} p_j \log(p_j)\), which is equivalent to \(\min \sum_{j=1}^{c} p_j \log(p_j)\). Take this regularized term to (5), leading to our BSSF
\[
\begin{aligned}
& \min_{Y,W} \frac{||Y - XW||_F^2}{\text{tr}(Y^T\tilde{S}Y)} + \gamma \sum_{j=1}^{c} p_j \log(p_j) \\
& \text{s.t.} \quad ||W||_2,0 = k, \quad Y \in \{0, 1\}^{n \times c}, \quad \sum_{j=1}^{c} Y_{ij} = 1 \\
& \quad p_j = \frac{\sum_{i=1}^{n} Y_{ij}}{n}.
\end{aligned}
\]

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 \(V\) and rewrite (6) as the following:
\[
\begin{aligned}
& \min_{Y,V,W} \frac{||Y - XW||_F^2}{\text{tr}(Y^T\tilde{S}Y)} + \gamma \sum_{j=1}^{c} p_j \log(p_j) \\
& \text{s.t.} \quad ||V||_2,0 = k, \quad W = V, \quad Y \in \{0, 1\}^{n \times c}, \quad \sum_{j=1}^{c} Y_{ij} = 1 \\
& \quad p_j = \frac{\sum_{i=1}^{n} Y_{ij}}{n}.
\end{aligned}
\]

Then, by introducing the Lagrange multiplies \(\Lambda \in \mathbb{R}^{d \times c}\) and \(\rho \in \mathbb{R}^c\), we obtain its augmented Lagrange function
\[
\begin{aligned}
& \mathcal{L} = \frac{||Y - XW||_F^2}{\text{tr}(Y^T\tilde{S}Y)} + \gamma \sum_{j=1}^{c} p_j \log(p_j) \\
& \quad + \text{tr}(\Lambda^T(W - V)) + \sum_{j=1}^{c} \rho_j \left( p_j - \frac{\sum_{i=1}^{n} Y_{ij}}{n} \right) \\
& \quad + \frac{\mu}{2} \left( ||W - V||_F^2 + \sum_{j=1}^{c} \left( p_j - \frac{\sum_{i=1}^{n} Y_{ij}}{n} \right)^2 \right). \quad (8)
\end{aligned}
\]

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

1) Optimizing \(W\): When fixing the other variables, we denote \(a = [1/(\text{tr}(Y^T\tilde{S}Y))]\) for simplicity. Then, we obtain the subproblem with \(W\)
\[
\begin{aligned}
& \min_{W} a||Y - XW||_F^2 + \text{tr}(\Lambda^T(W - V)) + \frac{\mu}{2} ||W - V||_F^2. \quad (9)
\end{aligned}
\]

Since it is an unconstraint quadratic programming, its closed-form solution can be easily obtained by setting its partial derivative w.r.t. \(W\) to zero
\[
W = \left( X^TX + \frac{\mu}{2a} I \right)^{-1} \left( X^TY - \frac{\Lambda - \mu V}{2a} \right). \quad (10)
\]

Unfortunately, the size of \(X^TX + (\mu/2a)I\) 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 \(X^TX\) 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 \(X\) as \(X = PEQ^T\) with \(P \in \mathbb{R}^{d \times n}, \Sigma \in \mathbb{R}^{n \times n}\), and \(Q \in \mathbb{R}^{d \times n}\), where \(P\) and \(Q\) are orthogonal and \(\Sigma\) is diagonal. Note that this SVD can be computed in advance and

Authorized licensed use limited to: Anhui University. Downloaded on July 06,2023 at 08:03:18 UTC from IEEE Xplore. Restrictions apply.
just needs to be computed once. Then, the SVD of \(X^TX\) is 
\(X^TX = Q\Sigma^2Q^T\). According to the Woodbury identity, we have 
\[
\left(X^TX + \frac{\mu}{2\alpha}I\right)^{-1} = \left(Q\Sigma^2Q^T + \frac{\mu}{2\alpha}I\right)^{-1} \\
= \frac{2a}{\mu} I - \frac{4a^2}{\mu^2} Q \left(\Sigma^{-2} + \frac{2a}{\mu} I\right)^{-1} Q^T \\
= \frac{2a}{\mu} I - \frac{4a^2}{\mu^2} Q \left(\Sigma^{-2} + \frac{2a}{\mu} I\right)^{-1} Q^T.
\]

Note that \(\Sigma^{-2} + (2a/\mu)I\) is a diagonal matrix and its inverse can be computed in \(O(n)\) time. Then, taking it back to (10), we have 
\[
W = \left(\frac{2a}{\mu} I - \frac{4a^2}{\mu^2} Q \left(\Sigma^{-2} + \frac{2a}{\mu} I\right)^{-1} Q^T\right) \left(X^TY - \frac{\Lambda - \mu V}{2a}\right) \\
= \frac{1}{\mu} \left(2aX^TY - \Lambda + \mu V\right) \\
- \frac{4a^2}{\mu^2} Q \left(\Sigma^{-2} + \frac{2a}{\mu} I\right)^{-1} \left(Q^T \left(X^TY - \frac{\Lambda - \mu V}{2a}\right)\right).
\]
(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 \(X^TX\) in \(O(n^d)\) time, and in each iteration, we compute (10) directly in \(O(ndc + d^2c)\), which is linear with \(n\).

2) Optimizing \(V\): When optimizing \(V\), we aim to solve the following subproblem:

\[
\min_V \left\| V - \left( W + \frac{\Lambda}{\mu} \right) \right\|_F^2 \\
\text{s.t. } \left\| V \right\|_{2,0} = k.
\]
(12)

Equation (12) also has a closed-form solution. We first let \(\mathbf{V} = \mathbf{W} + (\Lambda/\mu)\), and then we look for the \(k\) rows with the largest \(\ell_2\)-norm and set other rows to zeros.

3) Optimizing \(p\): When optimizing \(p\), we find that this can be decoupled into \(c\) independent subproblems. Consider the \(j\)th one, which has the following form:

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

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:

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

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

4) Optimizing \(Y\): When fixing other variables, we can rewrite (8) as

\[
\min_Y \frac{\|Y - XW\|_F^2}{\text{tr}(Y^T\hat{S}Y)} + \sum_{j=1}^c \rho_j \left( p_j - \frac{\sum_{i=1}^n Y_{ij}}{n} \right) \\
+ \frac{\mu}{2} \sum_{j=1}^c \left( p_j - \frac{\sum_{i=1}^n Y_{ij}}{n} \right)^2
\]

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 \(X^TX\) costs \(O(n^2d)\) time. Then, in ADMM algorithms, optimizing \(W\) costs \(O(ndc)\) time. The time complexity of optimizing \(V\) is \(O(\kappa d + c\kappa d)\). When optimizing \(p\), the time complexity is \(O(ln)\) time to compute \(p\). Since \(\hat{S}\) is often a sparse graph matrix, assuming in each row there are \(\kappa(\kappa)\) nonzero elements in \(\hat{S}\), computing \(tr(Y^T\hat{S}Y)\) costs \(O(c\kappa n)\) time. Updating \(Y\) costs \(O(ndc + \kappa n)\) time. Supposing the number of iterations is \(l\), the entire time complexity is \(O(n^2d + l(ndc + \kappa d + \kappa c + c\kappa n))\), which is linear with \(d\).

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

### Algorithm 1 BSFS

**Input:** Data matrix \(X\), normalized graph matrix \(\hat{S}\), parameter \(\gamma\) and number of selected features \(k\).

**Output:** Selected features.

1: Initialize \(Y\) by \(\max X^T\hat{S}Y\). Compute the SVD of \(X^TX\).
2: Initialize \(W\) by \(\min \|Y - XW\|_F^2\), initialize \(V = W\) and compute \(p_j = \sum_{i=1}^n Y_{ij}/n\), \(\Lambda = 0\), \(\rho = 0\) and \(\mu = 1\).
3: while not converge do
6: Compute \(W\) by Eq.(11).
7: Compute \(V\) by solving Eq.(12).
8: Compute \(p\) by solving Eq.(14).
9: Compute \(Y\) by solving Eq.(13).
10: Update the Lagrange multipliers by Eq.(16).
11: end while

9: Select the \(k\) features corresponding to the \(k\) non-zero rows of \(W\).

\[
s.t. \ Y \in \{0,1\}^{p \times c}, \ \sum_{j=1}^c Y_{ij} = 1. \tag{15}
\]
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; 4) Basehock; 5) USPS49 [67]; 6) Lung; 7) Leukemia; 8) Gisette; and 9) MNIST. The details of the datasets are summarized in Table I.

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

### 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.

#### 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 \( W \) 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.
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, ..., 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, ..., 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

Authorized licensed use limited to: Anhui University. Downloaded on July 06,2023 at 08:03:18 UTC from IEEE Xplore. Restrictions apply.
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

<table>
<thead>
<tr>
<th>Methods</th>
<th>metric</th>
<th>PIE</th>
<th>News4b</th>
<th>CBCL</th>
<th>Basehock</th>
<th>USPS49</th>
<th>Lung</th>
<th>Leukemia</th>
<th>Gisette</th>
<th>MNIST</th>
</tr>
</thead>
<tbody>
<tr>
<td>All features+ISC</td>
<td>ACC</td>
<td>0.3193</td>
<td>0.4584</td>
<td>0.5540</td>
<td>0.5665</td>
<td>0.7591</td>
<td>0.4932</td>
<td>0.6528</td>
<td>0.6403</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>NMI</td>
<td>0.5584</td>
<td>0.1253</td>
<td>0.0084</td>
<td>0.0128</td>
<td>0.2036</td>
<td>0.3950</td>
<td>0.0762</td>
<td>0.0576</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>NE</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>All features+BCLS</td>
<td>ACC</td>
<td>0.1015</td>
<td>0.2682</td>
<td>0.6275</td>
<td>0.5143</td>
<td>0.5146</td>
<td>0.2740</td>
<td>0.5833</td>
<td>0.5001</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>NMI</td>
<td>0.3256</td>
<td>0.0011</td>
<td>0.0474</td>
<td>0.0000</td>
<td>0.0001</td>
<td>0.1771</td>
<td>0.0097</td>
<td>0.0000</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>NE</td>
<td>0.9986</td>
<td>0.9995</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.9702</td>
<td>0.9726</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>All features+BCKM</td>
<td>ACC</td>
<td>0.3564</td>
<td>0.3343</td>
<td>0.5360</td>
<td>0.6116</td>
<td>0.7561</td>
<td>0.6164</td>
<td>0.6528</td>
<td>0.5111</td>
<td>0.4871</td>
</tr>
<tr>
<td></td>
<td>NMI</td>
<td>0.6734</td>
<td>0.0302</td>
<td>0.0037</td>
<td>0.0363</td>
<td>0.1986</td>
<td>0.6106</td>
<td>0.0762</td>
<td>0.0001</td>
<td>0.4271</td>
</tr>
<tr>
<td></td>
<td>NE</td>
<td>0.9975</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.9872</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>All features+BCM</td>
<td>ACC</td>
<td>0.3151</td>
<td>0.2956</td>
<td>0.5160</td>
<td>0.5855</td>
<td>0.5093</td>
<td>0.8356</td>
<td>0.6250</td>
<td>0.5000</td>
<td>0.1089</td>
</tr>
<tr>
<td></td>
<td>NMI</td>
<td>0.6546</td>
<td>0.0168</td>
<td>0.0001</td>
<td>0.0212</td>
<td>0.0000</td>
<td>0.7579</td>
<td>0.0380</td>
<td>0.0000</td>
<td>0.0211</td>
</tr>
<tr>
<td></td>
<td>NE</td>
<td>0.9461</td>
<td>0.3760</td>
<td>1</td>
<td>0.9994</td>
<td>0.0133</td>
<td>0.9441</td>
<td>0.9911</td>
<td>0.0000</td>
<td>0.0027</td>
</tr>
<tr>
<td>BSFS+kmeans</td>
<td>ACC</td>
<td>0.5075</td>
<td>0.2809</td>
<td>0.6952</td>
<td>0.6126</td>
<td>0.7923</td>
<td>0.7027</td>
<td>0.7153</td>
<td>0.8357</td>
<td>0.5292</td>
</tr>
<tr>
<td></td>
<td>NMI</td>
<td>0.8040</td>
<td>0.0339</td>
<td>0.1134</td>
<td>0.0408</td>
<td>0.2667</td>
<td>0.6597</td>
<td>0.1285</td>
<td>0.3578</td>
<td>0.5030</td>
</tr>
<tr>
<td></td>
<td>NE</td>
<td>0.9698</td>
<td>0.1577</td>
<td>0.9977</td>
<td>0.9450</td>
<td>0.9880</td>
<td>0.9400</td>
<td>0.9993</td>
<td>0.9638</td>
<td>0.9819</td>
</tr>
</tbody>
</table>

as strong as those balanced clustering. Despite this, the NE of our method is close 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
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 ($\gamma$), which is needed to be tuned manually. To explore the effect of $\gamma$ 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 $\gamma$. Therefore, the selection of the parameter $\gamma$ 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 $\gamma$. (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 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


![Peng Zhou](image1.png) 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](image2.png) 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](image3.png) 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](image4.png) 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.