Knowledge-Based Systems 193 (2020) 105417

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Unsupervised feature selection for balanced clustering☆

Peng Zhou [a,b,*], Jiangyong Chen [a], Mingyu Fan [c], Liang Du [d], Yi-Dong Shen [b], Xuejun Li [a]

[a] School of Computer Science and Technology, Anhui University, Hefei 230601, China
[b] The State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing 100190, China
[c] College of Maths and Information Science, Wenzhou University, Wenzhou 325035, China
[d] School of Computer and Information Technology, Shanxi University, Taiyuan 030006, China

ARTICLE INFO

Article history:
Received 13 July 2019
Received in revised form 30 November 2019
Accepted 19 December 2019
Available online 24 December 2019

Keywords:
Feature selection
Balanced clustering
ADMM

ABSTRACT

In many real-world applications of data mining, such as energy load balance of wireless sensor networks, given data points with balanced distribution, i.e., each class contains approximately the same number of instances, we often need to obtain a clustering result to reflect such balance. In many data, especially the high-dimensional data, such balanced structure is not obvious in the original feature space, due to the noisy and redundant features. Therefore we need to apply feature selection methods to pick several informative features to reveal such balanced structure of data. Feature selection is a fundamental problem in machine learning tasks and has attracted considerable attentions in recent years. However, conventional feature selection methods often focus on how to select the most discriminative features, whereas ignoring the balance property of the data. To tackle this problem, we propose a novel unsupervised feature selection method for balanced clustering which can reveal the intrinsic balanced structure of data. In our method, a balanced regularization term is introduced to select the features which can help to produce balanced clusters. Then, we provide an Alternating Direction Method of Multipliers (ADMM) to optimize the introduced objective function. At last, the experiments are conducted on six benchmark data sets, including Yale and 20NG data sets and so on, by comparing with other state-of-the-art unsupervised feature selection methods published in the literature. The experimental results show that our method not only has better clustering performance but also leads to more balanced clustering structure.

1. Introduction

In many real-world data mining applications, given instances with balanced distribution, i.e., each class contains approximately the same number of instances, it is quite often to require to generate the balanced clusters. A good clustering method should prevent a too small or too great number of data points from being partitioned into a cluster. For example, in photo query systems, we often need to organize the photos in a balanced way because such balanced layout can help the users to orient and find specific photos more efficiently [1]. Another scenario for balanced clustering is energy load balance of wireless sensor networks, where the unbalance of cluster structure may lead to the unbalance of energy consumption and may shorten the network lifetime [2]. Moreover, according to the results in [3], balanced clustering tends to avoid generating outlier clusters, and thus may obtain a better clustering performance. Therefore, to characterize the balanced structure and generate balanced clustering is quite important and essential in many applications.

However, this balanced structure of the data, especially the high-dimensional data, may not be so obvious in the original feature space due to the noisy and redundant features. To better reveal the balanced structure of data, we need to select such informative features. Feature selection is a fundamental problem in machine learning and data mining tasks, and has been widely studied [4–11]. These methods focus on how to select the discriminative features and discard the redundant ones to obtain a better performance on classification or clustering. For example, Yang et al. applied local total scatter and between-class scatter matrix to evaluate features [12]; Zhu et al. proposed a co-regularized feature selection method using the heat kernel to construct the similarity matrix [13]; Luo et al. constructed

☆ No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.knosys.2019.105417.

* Corresponding author at: School of Computer Science and Technology, Anhui University, Hefei 230601, China.
E-mail addresses: zhoupeng@ahu.edu.cn (P. Zhou), e18301199@stu.ahu.edu.cn (J. Chen), fanmingyu@amss.ac.cn (M. Fan), duliang@sxu.edu.cn (L. Du), ydshen@ios.ac.cn (Y.-D. Shen), xjli@ahu.edu.cn (X. Li).

https://doi.org/10.1016/j.knosys.2019.105417
0950-7051/

the adaptive graph with structure regularization to select features [10]. However, they do not pay any attention on revealing the balanced structure of data. This problem is even worse in unsupervised learning because we are lack of the guides of labels.

To tackle this problem, we propose a novel Feature Selection method for Balanced Clustering (FSBC). In our method, we select not only the discriminative features but also the ones which can preserve the balanced structure. To reveal the balanced structure, we need to obtain the clusters of data. Since k-means is one of the most popular clustering algorithm, our method is in a k-means framework. By applying k-means, we can obtain the clustering result, and we propose a balanced regularization term on such clustering result to guide us to select the appropriate features. We integrate the balanced k-means clustering and feature selection in a unified framework. On one hand, balanced k-means can reveal the balanced clustering structure and guide us to select the features to preserve such structure; and on the other hand, the selected features can help us obtain a more accurate clustering result. Therefore, we jointly do clustering and select features so that the two tasks can be boosted by each other. Moreover, when selecting the features, different from the traditional feature selection methods which using $\ell_{2,1}$-norm or $\ell_{2,p}$-norm to sort the features, we apply the $\ell_{2,0}$-norm directly to pick the exact top-k features. Since the introduced objective function is non-convex and discontinuous, it is hard to be optimized. To address this issue, we provide an effective Alternating Direction Method of Multipliers (ADMM) [14] to optimize it. At last, we conduct experiments on benchmark data sets and the experimental results demonstrate the effectiveness of our method.

It is worthy to highlight the main contributions of our paper here:

- To the best of our knowledge, we are the first to propose an unsupervised feature selection method for balanced clustering. Different from the existing unsupervised feature selection methods which only focus on the informative features, our method not only picks the informative features but also select those which can reveal the balanced structure of data.
- We integrate the balanced k-means and feature selection into a unified framework and propose an effective ADMM algorithm to jointly do clustering and select features.
- We conduct the experiments on six benchmark data sets, such as Yale and 20NG, and compare our method with the state-of-the-art unsupervised feature selection methods. The experimental results show that our method outperforms the compared state-of-the-art unsupervised feature selection not only on the accuracy but also the balance of the clustering results.

The paper is organized as follows. Section 2 describes some related work. Section 3 presents our unsupervised feature selection method in details. Section 4 shows the experimental results, and Section 5 concludes the paper.

2. Related work

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

2.1. Feature selection

According to the availability of labels of data, feature selection methods can be roughly classified into three categorizes: supervised feature selection, semi-supervised feature selection and unsupervised feature selection.

By exploiting the label information, supervised feature selection is usually able to identify discriminative features for classification [15]. For example, Gu et al. selected features based on the generalized Fisher score of each feature [16]; Nie et al. proposed a robust feature selection method with $\ell_{2,1}$ regression [6]; Fan et al. utilized $\ell_{2,0}$-norm to select exact top-k features [17].

With insufficient class labels, semi-supervised methods are proposed to propagate the label information and select the features which consider both the label information and the intrinsic structure of data. For example, Han et al. utilized the manifold structure of data for semi-supervised feature selection [18]; Chang et al. proposed a semi-supervised feature selection for multi-label data [19]; furthermore, Chang et al. also provided a semi-supervised feature selection for multi-task learning [20]; most recently, Luo et al. presented a semi-supervised feature selection method with adaptive neighbor assignment [21].

In unsupervised learning, due to the absence of class labels, feature selection is a more challenging problem. It tries to select features which can well preserve the intrinsic structure of the data. According to the different intrinsic structure which are captured, various unsupervised feature selection methods are proposed. For example, some methods try to reconstruct the original data well by the selected features, such as [22].

Some approaches aim to preserve the pseudo labels, for example, Hou et al. integrated manifold embedding and feature selection into a general framework by preserving the pseudo labels [23]; Wang et al. selected features to preserve the pseudo labels generated by matrix factorization [24]; Shi et al. proposed a robust unsupervised feature selection method to preserve the pseudo labels [25]. These methods jointly learn the pseudo labels and selected the features, so that the two tasks can be boosted by each other.

Some methods focus on some special type of data, e.g. text data, and try to preserve some important information of the original data. For example, to handle text data, Abualigah et al. proposed several particle swarm optimization algorithms to select useful features for each document and applied them to document clustering [26,27]. In text data, words are the natural features. These methods select the informative words to reveal the topic of each document and have been demonstrated promising performance.

Subspace clustering is a kind of famous unsupervised learning methods, so some works extend it to feature selection tasks and select features to preserve the subspace structure. Fan et al. proposed a discriminative subspace clustering model for feature selection which can preserve both the hard and soft structure of data [28]; Zheng et al. used sparse subspace learning to select features [29]. These methods learn the subspace structure of data in the process of feature selection so that the selected features can preserve such structure well.

As a very important structure of data, graph is also often preserved in feature selection methods. Du et al. learned an adaptive graph for feature selection by preserving the global and local structure [30]; Nie et al. adaptively learned the local structure from the results of feature selection [31]; Li et al. proposed a generalized uncorrelated regression with adaptive graph for feature selection [32]. These methods construct an adaptive graph in the procedure of feature selection, i.e., the graph changes with selected features. By this way, graph learning and feature selection can also be boosted by each other.

Note that none of the above methods considered the balanced structure of data, so that they may be inappropriate to the scenarios which require to generate the balanced clusters of data. This problem may be worse in unsupervised learning

because of the absence of the label information. To tackle this problem, we propose a novel unsupervised feature selection for balanced clustering in this paper. Note that in our method we select features for not only revealing the balanced structure, but also preserving the pseudo labels to obtain a better clustering performance.

2.2. Balanced clustering

Clustering is a fundamental problem in unsupervised learning. Although it has been widely studied [33–37], few of them pay attention on the balanced structure of data. In recent years, some balanced clustering methods have been proposed to handle the data with balanced distribution. The balanced clustering algorithms can be roughly categorized into two types: hard-balanced clustering and soft-balanced clustering.

In the hard-balanced clustering, the cluster size is strictly set as a fixed number. For example, Bradley et al. proposed a constrained k-means method which takes the cluster sizes as parameters [38]. Then, Malinen et al. provided a balanced k-means method which also imposed a balanced constraint on k-means [39]. Costa et al. imposed the balanced constraint on the minimum sum-of-squares clustering leading to a balanced minimum sum-of-squares clustering method [40].

In many applications, the absolute balance is often not required, thus some soft-balanced clustering methods are proposed. In this kind of methods, the balance is an aim but not a mandatory requirement. For example, Banerjee et al. applied the balance as a penalty term to make the clustering result balanced [41,42]. Zhong et al. proposed a model-based clustering with soft balancing [3]. Liu et al. used the exclusive lasso as the balanced regularized term and imposed it on the least square regression for clustering [43]. Li et al. applied the exclusive lasso to the k-means and min-cut leading to balanced k-means and balanced min-cut methods [44]. In this paper, we focus on this soft-balanced clustering which does not require the strict balance.

3. Unsupervised feature selection for balanced clustering

In this section, we will introduce our feature selection method in details. Throughout this paper, we use boldface uppercase and lowercase letters to denote matrices and vectors, respectively. The (i,j)th element of a matrix $\mathbf{M}$ is denoted as M_{ij} and the ith element of a vector $\mathbf{v}$ is denoted as v_i. We use $\mathbf{M}_{i\cdot}$ and $\mathbf{M}_{\cdot i}$ to denote the ith row and the ith column of matrix $\mathbf{M}$, respectively. diag($\mathbf{v}$) ($\mathbf{v} \in \mathbb{R}^d$) is a diagonal matrix whose diagonal elements are the entries of vector $\mathbf{v}$ and diag($\mathbf{M}$) ($\mathbf{M} \in \mathbb{R}^{d\times d}$) is a d-dimensional vector consists of the diagonal elements of the matrix $\mathbf{M}$. We denote the $\ell_{2,0}$-norm of $\mathbf{M}$ as $\|\mathbf{M}\|_{2,0}$, which means the number of non-zero columns in the $\mathbf{M}$ and denote the ℓ_0-norm of the vector $\mathbf{v}$ as $\|\mathbf{v}\|_0$, which means the number of non-zero elements in $\mathbf{v}$. Since $\ell_{2,0}$-norm is non-convex and discontinuous, $\ell_{2,1}$-norm is often used as an approximation of $\ell_{2,0}$-norm. $\ell_{2,1}$-norm of $\mathbf{M} \in \mathbb{R}^{n\times d}$ is defined as $\sum_{j=1}^{d}\sqrt{\sum_{i=1}^{n} M_{ij}^2}$.

The basic idea of our method is that we jointly do balanced clustering and select features. Since we aim to select the features which can reveal the balanced structure, we need to apply the balanced clustering to obtain such structure and use it to guide the feature selection. When we select some features, we can also find a clearer balanced structure by doing balanced clustering with the selected features. Therefore, the balanced clustering and feature selection can be boosted by each other with this joint learning framework.

3.1. Formulation

As introduced before, our method applies k-means to generate the clustering result. In k-means, given a data matrix $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^{d\times n}$, where $\mathbf{x}_i \in \mathbb{R}^d$ is an instance in the data set, we need to optimize the following objective function to obtain the clustering result:

$$\begin{aligned} \min_{\mathbf{G},\mathbf{F}} \quad & \|\mathbf{X} - \mathbf{G}\mathbf{F}^T\|_F^2 \\ s.t. \quad & \mathbf{F} \in \{0,1\}^{n\times c}, \quad \sum_{m=1}^{c} F_{im} = 1. \end{aligned} \tag{1}$$

where c is the number of clusters, $\mathbf{G} \in \mathbb{R}^{d\times c}$ is the cluster centroid matrix, $\mathbf{F}$ is the clustering indicator matrix, i.e., $F_{im} = 1$ if $\mathbf{x}_i$ belongs to the mth cluster and $F_{im} = 0$ otherwise.

Note that, each column $\mathbf{F}_{\cdot m}$ in $\mathbf{F}$ denotes the instances in the mth cluster. Therefore, we can obtain the number of instances in the mth cluster, denoted as n_m, by summing up $\mathbf{F}_{\cdot m}$, i.e., $n_m = \sum_{i=1}^{n} F_{im}$. Furthermore, we can obtain the distribution $\mathbf{p} \in \mathbb{R}^c$ of the number of instances in each cluster by:

$$p_m = \frac{n_m}{\sum_{m=1}^{c} n_m} = \frac{n_m}{n}. \tag{2}$$

Since we wish the clustering result should be balanced, i.e., each cluster has approximately the same number of instances, we can minimize the negative Shannon Entropy of the distribution $\mathbf{p}$. Therefore, we use the negative Shannon Entropy of the distribution $\mathbf{p}$, i.e., $\sum_{m=1}^{c} p_m \log(p_m)$, as the balanced regularization term, and take it into Eq. (1), leading to the balanced k-means:

$$\begin{aligned} \min_{\mathbf{G},\mathbf{F}} \quad & \|\mathbf{X} - \mathbf{G}\mathbf{F}^T\|_F^2 + \lambda \sum_{m=1}^{c} p_m \log(p_m) \\ s.t. \quad & \mathbf{F} \in \{0,1\}^{n\times c}, \quad \sum_{m=1}^{c} F_{im} = 1, \\ & p_m = \frac{\sum_{i=1}^{n} F_{im}}{n}. \end{aligned} \tag{3}$$

where λ is a hyper-parameter.

In order to select the features to preserve such balanced structure, we introduce a projection matrix $\mathbf{W} \in \mathbb{R}^{c\times d}$ to map the data from original feature space to a new low-dimensional feature space by $\mathbf{WX}$. Intuitively, since we need to select the top-k features, we wish the number of non-zero columns in $\mathbf{W}$ is just k, thus we need an $\ell_{2,0}$-norm constraint on $\mathbf{W}$. Therefore, we obtain the following formulation:

$$\begin{aligned} \min_{\mathbf{G},\mathbf{F},\mathbf{W}} \quad & \|\mathbf{W}\mathbf{X} - \mathbf{G}\mathbf{F}^T\|_F^2 + \lambda \sum_{m=1}^{c} p_m \log(p_m) + \tau \|\mathbf{W}\|_F^2 \\ s.t. \quad & \mathbf{F} \in \{0,1\}^{n\times c}, \quad \sum_{m=1}^{c} F_{im} = 1, \\ & p_m = \frac{\sum_{i=1}^{n} F_{im}}{n}, \\ & \|\mathbf{W}\|_{2,0} = k. \end{aligned} \tag{4}$$

where $\|\mathbf{W}\|_F^2$ is a Frobenius norm regularized term on $\mathbf{W}$ as the prior and τ is another hyper-parameter. Note that, different from conventional feature selection methods [6,12,30], which use $\ell_{2,1}$-norm or $\ell_{2,p}$-norm to approximate the $\ell_{2,0}$-norm, we apply $\ell_{2,0}$-norm directly. The conventional methods need to score every features under some criterion and select the k top features according to the scores. However, in our method, we impose the $\ell_{2,0}$-norm on $\mathbf{W}$ without any approximation, so that we can directly select the exact top-k features, which is more desirable than using $\ell_{2,1}$-norm or $\ell_{2,p}$-norm.

3.2. Optimization

Since Eq. (4) is non-convex and discontinuous, the optimization is difficult. To handle this problem, we propose an ADMM method to optimize it.

Firstly, for the convenience of the optimization, we can further transform the $\ell_{2,0}$-norm in Eq. (4) into ℓ_0-norm by introducing an indicator vector $\mathbf{v} \in \{0,1\}^d$ of features. $v_i = 1$ indicates that the ith feature should be selected and $v_i = 0$ means it should not be selected. Then we can reformulate Eq. (4) as follows:

$$\begin{aligned} \min_{\mathbf{G},\mathbf{F},\mathbf{W},\mathbf{v}} \quad & \|\mathbf{W}\mathrm{diag}(\mathbf{v})\mathbf{X} - \mathbf{G}\mathbf{F}^T\|_F^2 + \lambda \sum_{m=1}^{c} p_m \log(p_m) + \tau \|\mathbf{W}\|_F^2 \\ s.t. \quad & \mathbf{F} \in \{0,1\}^{n\times c}, \quad \sum_{m=1}^{c} F_{im} = 1, \\ & p_m = \frac{\sum_{i=1}^{n} F_{im}}{n}, \\ & \mathbf{v} \in \{0,1\}^d, \quad \sum_{i=1}^{d} v_i = k. \end{aligned} \tag{5}$$

In Eq. (5), we simultaneously do balanced k-means (learning $\mathbf{G}$ and $\mathbf{F}$) and select features (learning $\mathbf{W}$ and $\mathbf{v}$), so that the two tasks can be boosted by each other.

By introducing the Lagrange multipliers γ_m ($m = 1, \ldots, c$), we obtain the augmented Lagrange function of Eq. (5):

$$\begin{aligned} \min_{\mathbf{G},\mathbf{F},\mathbf{W},\mathbf{v},\mathbf{p}} \quad & \mathcal{L}_1 = \|\mathbf{W}\mathrm{diag}(\mathbf{v})\mathbf{X} - \mathbf{G}\mathbf{F}^T\|_F^2 \\ & + \lambda \sum_{k=1}^{c} p_m \log(p_m) + \tau \|\mathbf{W}\|_F^2 \\ & + \sum_{m=1}^{c} \gamma_m \left(p_m - \frac{\sum_{i=1}^{n} F_{im}}{n} \right) \\ & + \frac{\mu_1}{2} \sum_{m=1}^{c} \left(p_m - \frac{\sum_{i=1}^{n} F_{im}}{n} \right)^2, \\ s.t. \quad & \mathbf{F} \in \{0,1\}^{n\times c}, \quad \sum_{m=1}^{c} F_{im} = 1, \\ & \mathbf{v} \in \{0,1\}^d, \quad \sum_{i=1}^{d} v_i = k. \end{aligned} \tag{6}$$

where μ_1 is an adaptive parameter.

Then we optimize one variable while fixing the others.

3.2.1. Optimizing $\mathbf{W}$

At beginning, we initialize $\mathbf{W}$ by doing Principal Component Analysis (PCA) [45] on $\mathbf{X}$. In the following iterations, when fixing the other variables, we can rewrite Eq. (6) as:

$$\min_{\mathbf{W}} \quad \|\mathbf{W}\mathrm{diag}(\mathbf{v})\mathbf{X} - \mathbf{G}\mathbf{F}^T\|_F^2 + \tau \|\mathbf{W}\|_F^2. \tag{7}$$

Obviously, Eq. (7) is similar with ridge regression and has a closed-form solution by setting the partial derivative of Eq. (7) w.r.t. $\mathbf{W}$ to zero:

$$\mathbf{W} = \mathbf{G}\mathbf{F}^T\mathbf{X}^T\mathrm{diag}(\mathbf{v}) \left(\mathrm{diag}(\mathbf{v})\mathbf{X}\mathbf{X}^T\mathrm{diag}(\mathbf{v}) + \tau\mathbf{I} \right)^{-1} \tag{8}$$

where $\mathbf{I}$ is the identity matrix.

Note that $\mathrm{diag}(\mathbf{v})\mathbf{X}\mathbf{X}^T\mathrm{diag}(\mathbf{v}) + \tau\mathbf{I}$ is a d-by-d matrix and it costs $O(d^3)$ time to compute its inverse. Fortunately, since $\mathbf{v}$ only contains k ones, $\mathrm{diag}(\mathbf{v})\mathbf{X}\mathbf{X}^T\mathrm{diag}(\mathbf{v})$ only contains k non-zero columns and rows and we can compute its inverse very efficiently.

In more details, we extract the $k \times k$ non-zero principal minor of $\mathrm{diag}(\mathbf{v})\mathbf{X}\mathbf{X}^T\mathrm{diag}(\mathbf{v})$ as $\mathbf{B} \in \mathbb{R}^{k\times k}$. Then we extend it to $\mathbf{C} \in \mathbb{R}^{d\times d}$:

$$\mathbf{C} = \begin{pmatrix} \mathbf{B} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \tag{9}$$

It is obvious that $\mathbf{C}$ can be obtained by exchanging several rows and columns of $\mathrm{diag}(\mathbf{v})\mathbf{X}\mathbf{X}^T\mathrm{diag}(\mathbf{v})$ simultaneously. More formally, by defining the permutation matrix $\mathbf{P} \in \{0,1\}^{d\times d}$, we can obtain $\mathbf{C}$ by $\mathbf{C} = \mathbf{P}^T\mathrm{diag}(\mathbf{v})\mathbf{X}\mathbf{X}^T\mathrm{diag}(\mathbf{v})\mathbf{P}$.

Now, considering $(\mathbf{C} + \tau\mathbf{I})^{-1}$, we have

$$(\mathbf{C} + \tau\mathbf{I})^{-1} = \begin{pmatrix} \mathbf{B} + \tau\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \tau\mathbf{I} \end{pmatrix}^{-1} = \begin{pmatrix} (\mathbf{B} + \tau\mathbf{I})^{-1} & \mathbf{0} \\ \mathbf{0} & \frac{1}{\tau}\mathbf{I} \end{pmatrix} \tag{10}$$

Note that $\mathbf{B}+\tau\mathbf{I}$ is a k-by-k matrix whose inverse can be computed in $O(k^3)$ time and $k \ll d$. Therefore, we takes $O(k^3)$ time to compute $(\mathbf{C} + \tau\mathbf{I})^{-1}$, and then obtain $\left(\mathrm{diag}(\mathbf{v})\mathbf{X}\mathbf{X}^T\mathrm{diag}(\mathbf{v}) + \tau\mathbf{I}\right)^{-1}$ by

$$\left(\mathrm{diag}(\mathbf{v})\mathbf{X}\mathbf{X}^T\mathrm{diag}(\mathbf{v}) + \tau\mathbf{I}\right)^{-1} = \left(\mathbf{P}\mathbf{C}\mathbf{P}^T + \tau\mathbf{I}\right)^{-1} = \mathbf{P}(\mathbf{C} + \tau\mathbf{I})^{-1}\mathbf{P}^T \tag{11}$$

To sum up, we take $O(nk^2)$ time to compute $\mathrm{diag}(\mathbf{v})\mathbf{X}\mathbf{X}^T\mathrm{diag}(\mathbf{v})$, and $O(k^3)$ time to compute $(\mathbf{C} + \tau\mathbf{I})^{-1}$, thus we takes $O(nk^2 + k^3)$ time to compute $\left(\mathrm{diag}(\mathbf{v})\mathbf{X}\mathbf{X}^T\mathrm{diag}(\mathbf{v}) + \tau\mathbf{I}\right)^{-1}$. After that, we compute $\mathbf{W}$ by a series of matrix multiplications, which cost $O(ndk + nck + dck)$ time. Therefore, the whole time complexity is $O(ndk + nck + dck + nk^2 + k^3)$.

3.2.2. Optimizing $\mathbf{v}$

When optimizing $\mathbf{v}$, we obtain the following subproblem:

$$\begin{aligned} \min_{\mathbf{v}} \quad & \|\mathbf{W}\mathrm{diag}(\mathbf{v})\mathbf{X} - \mathbf{G}\mathbf{F}^T\|_F^2, \\ s.t. \quad & \mathbf{v} \in \{0,1\}^d, \quad \sum_{i=1}^{d} v_i = k. \end{aligned} \tag{12}$$

Eq. (12) is a 0–1 integer programming and is generally difficult to solve. Here we utilize the ℓ_2-box method [17,46] to solve this problem. According to [17,46], the binary constraint can be replaced with an equivalent set of continuous constraint, i.e., the intersection of a box and a shifted ℓ_2-sphere. It is presented in the following Theorem:

Theorem 1 ([17,46]). *Let* $\mathbf{1}$ *be the vector whose entries are all 1s, we have*

$$\mathbf{v} \in \{0,1\}^d \Leftrightarrow \left\{\mathbf{v} : \mathbf{v} \in [0,1]^d\right\} \bigcap \left\{\mathbf{v} : \left\|\mathbf{v} - \frac{\mathbf{1}}{2}\right\|_2^2 = \frac{d}{4}\right\}.$$

According to this Theorem, we can involve two auxiliary variables $\mathbf{v}_1, \mathbf{v}_2 \in \mathbb{R}^d$, where $\mathbf{v}_1$ is in a box, i.e., $\mathbf{v}_1 \in S_b$ and $S_b = \{\mathbf{x} : \mathbf{x} \in [0,1]^d\}$, and $\mathbf{v}_2$ is in a shifted ℓ_2-sphere, i.e., $\mathbf{v}_2 \in S_p$ and $S_p = \left\{\mathbf{x} : \left\|\mathbf{x} - \frac{\mathbf{1}}{2}\right\|_2^2 = \frac{d}{4}\right\}$. Then we can obtain the following equivalent formulation:

$$\begin{aligned} \min_{\mathbf{v}} \quad & \|\mathbf{W}\mathrm{diag}(\mathbf{v})\mathbf{X} - \mathbf{G}\mathbf{F}^T\|_F^2, \\ s.t. \quad & \sum_{i=1}^{d} v_i = k, \quad \mathbf{v} = \mathbf{v}_1, \quad \mathbf{v}_1 \in S_b, \quad \mathbf{v} = \mathbf{v}_2, \quad \mathbf{v}_2 \in S_p. \end{aligned} \tag{13}$$

Eq. (13) can also be solved by ADMM. By introducing the Lagrange multipliers $\mathbf{y}_1 \in \mathbb{R}^d$, $\mathbf{y}_2 \in \mathbb{R}^d$ and y_3, we obtain its augmented Lagrange function:

$$\mathcal{L}_2 = \|\mathbf{W}\mathrm{diag}(\mathbf{v})\mathbf{X} - \mathbf{G}\mathbf{F}^T\|_F^2 + \mathbf{y}_1^T(\mathbf{v} - \mathbf{v}_1) + \mathbf{y}_2^T(\mathbf{v} - \mathbf{v}_2)$$

$$+ y_3(\mathbf{1}^T\mathbf{v} - k) + \frac{\mu_2}{2}\left(\|\mathbf{v}-\mathbf{v}_1\|_2^2 + \|\mathbf{v}-\mathbf{v}_2\|_2^2 + (\mathbf{1}^T\mathbf{v}-k)^2\right) \tag{14}$$

When optimizing $\mathbf{v}$ by fixing $\mathbf{v}_1$ and $\mathbf{v}_2$, it is an unconstraint quadratic optimization problem and can be solved by setting its partial derivative w.r.t. $\mathbf{v}$ to zero. Its closed-form solution is:

$$\mathbf{v} = \left(2(\mathbf{W}^T\mathbf{W}) \odot (\mathbf{X}\mathbf{X}^T) + \mu_2(2\mathbf{I} + \mathbf{1}\mathbf{1}^T)\right)^{-1}\left(2\mathrm{diag}(\mathbf{X}\mathbf{F}\mathbf{G}^T\mathbf{W}) - \mathbf{y}_1 - \mathbf{y}_2 + \mu_2(\mathbf{v}_1+\mathbf{v}_2) + (y_3 - \rho k)\mathbf{1}\right) \tag{15}$$

where $\odot$ is Hadamard product, which means the element-wise production of two matrices.

Then, by fixing $\mathbf{v}$, we update $\mathbf{v}_1$ and $\mathbf{v}_2$ by projecting $\mathbf{v} + \frac{\mathbf{y}_1}{\rho}$ and $\mathbf{v} + \frac{\mathbf{y}_2}{\rho}$ into S_b and S_p, respectively:

$$\begin{cases} \mathbf{v}_1 = P_{S_b}\left(\mathbf{v} + \frac{\mathbf{y}_1}{\rho}\right) \\ \mathbf{v}_2 = P_{S_p}\left(\mathbf{v} + \frac{\mathbf{y}_2}{\rho}\right) \end{cases} \tag{16}$$

For any x, P_{S_b} is an element-wise function, which is defined as:

$$P_{S_b}(x) = \begin{cases} 0, & \text{if } x < 0 \\ x, & \text{if } 0 \le x \le 1 \\ 1, & \text{otherwise.} \end{cases} \tag{17}$$

Given a vector $\mathbf{x} \in \mathbb{R}^d$, P_{S_p} can be computed as follows:

$$P_{S_p}(\mathbf{x}) = \frac{\mathbf{1}}{2} + \left(\frac{\sqrt{d}}{2\|\mathbf{x} - \frac{\mathbf{1}}{2}\|_2}\right)\left(\mathbf{x} - \frac{\mathbf{1}}{2}\right) \tag{18}$$

At last, we update the Lagrange multipliers as follows:

$$\begin{cases} \mathbf{y}_1 = \mathbf{y}_1 + \rho(\mathbf{v} - \mathbf{v}_1) \\ \mathbf{y}_2 = \mathbf{y}_2 + \rho(\mathbf{v} - \mathbf{v}_2) \\ y_3 = y_3 + \rho(\mathbf{1}^T\mathbf{v} - k) \\ \mu_2 = \alpha\mu_2. \end{cases} \tag{19}$$

where $\alpha > 1$ is a given parameter.

It is obvious that the most expensive step of this ADMM is to compute $\mathbf{v}$ by Eq. (15). $2(\mathbf{W}^T\mathbf{W}) \odot (\mathbf{X}\mathbf{X}^T)$ and $2\mathrm{diag}(\mathbf{X}\mathbf{F}\mathbf{G}^T\mathbf{W})$ only need to be computed once outside the iteration and costs $O(cd^2 + nd^2)$ and $O(ndc + c^2d)$ time, respectively. However, in each iteration, we need to compute the inverse of $2(\mathbf{W}^T\mathbf{W}) \odot (\mathbf{X}\mathbf{X}^T) + \mu_2(2\mathbf{I} + \mathbf{1}\mathbf{1}^T)$ which costs $O(d^3)$ time.

To speedup this operation, we apply an incremental method to compute its inverse. Outside the iteration, we can compute $2(\mathbf{W}^T\mathbf{W}) \odot (\mathbf{X}\mathbf{X}^T)$ first, and then calculate its singular value decomposition (SVD) as:

$$2(\mathbf{W}^T\mathbf{W}) \odot (\mathbf{X}\mathbf{X}^T) = \mathbf{U}\mathbf{S}\mathbf{U}^T \tag{20}$$

where $\mathbf{U} \in \mathbb{R}^{d\times d}$ is the singular vector matrix and $\mathbf{S}$ is a diagonal matrix whose diagonal elements are the singular values of $2(\mathbf{W}^T\mathbf{W}) \odot (\mathbf{X}\mathbf{X}^T)$. The time complexity is $O(d^3)$, but we just need to compute it once.

Then, in each iteration, we incrementally compute the inverse of $2(\mathbf{W}^T\mathbf{W}) \odot (\mathbf{X}\mathbf{X}^T) + \mu_2(2\mathbf{I} + \mathbf{1}\mathbf{1}^T)$. Firstly, we compute $\left(2(\mathbf{W}^T\mathbf{W}) \odot (\mathbf{X}\mathbf{X}^T) + 2\mu_2\mathbf{I}\right)^{-1}$. Since we already have the SVD of $2(\mathbf{W}^T\mathbf{W}) \odot (\mathbf{X}\mathbf{X}^T)$, we can compute its inverse easily:

$$\left(2(\mathbf{W}^T\mathbf{W}) \odot (\mathbf{X}\mathbf{X}^T) + 2\mu_2\mathbf{I}\right)^{-1} = \mathbf{U}(\mathbf{S} + \frac{1}{2\mu_2}\mathbf{I})^{-1}\mathbf{U}^T \tag{21}$$

Next, according to Sherman–Morrison Equality, we have

$$\left(2(\mathbf{W}^T\mathbf{W}) \odot (\mathbf{X}\mathbf{X}^T) + 2\mu_2\mathbf{I} + \mu_2\mathbf{1}\mathbf{1}^T\right)^{-1} = \mathbf{U}(\mathbf{S} + \frac{1}{2\mu_2}\mathbf{I})^{-1}\mathbf{U}^T - \frac{\mu_2\mathbf{U}(\mathbf{S} + \frac{1}{2\mu_2}\mathbf{I})^{-1}\mathbf{U}^T\mathbf{1}\mathbf{1}^T\mathbf{U}(\mathbf{S} + \frac{1}{2\mu_2}\mathbf{I})^{-1}\mathbf{U}^T}{1 + \mu_2\mathbf{1}^T\mathbf{U}(\mathbf{S} + \frac{1}{2\mu_2}\mathbf{I})^{-1}\mathbf{U}^T\mathbf{1}} \tag{22}$$

Let $\mathbf{u} = \left(2\mathrm{diag}(\mathbf{X}\mathbf{F}\mathbf{G}^T\mathbf{W}) - \mathbf{y}_1 - \mathbf{y}_2 + \mu_2(\mathbf{v}_1 + \mathbf{v}_2) + (y_3 - \rho k)\mathbf{1}\right)$, we can compute $\mathbf{v}$ by

$$\begin{aligned} \mathbf{v} &= \left(2(\mathbf{W}^T\mathbf{W}) \odot (\mathbf{X}\mathbf{X}^T) + 2\mu_2\mathbf{I} + \mu_2\mathbf{1}\mathbf{1}^T\right)^{-1}\mathbf{u} \\ &= \mathbf{U}(\mathbf{S} + \frac{1}{2\mu_2}\mathbf{I})^{-1}\mathbf{U}^T\mathbf{u} \\ &\quad - \frac{\mu_2\mathbf{U}(\mathbf{S} + \frac{1}{2\mu_2}\mathbf{I})^{-1}\mathbf{U}^T\mathbf{1}\mathbf{1}^T\mathbf{U}(\mathbf{S} + \frac{1}{2\mu_2}\mathbf{I})^{-1}\mathbf{U}^T\mathbf{u}}{1 + \mu_2\mathbf{1}^T\mathbf{U}(\mathbf{S} + \frac{1}{2\mu_2}\mathbf{I})^{-1}\mathbf{U}^T\mathbf{1}} \end{aligned} \tag{23}$$

It costs $O(d^2)$ to compute $\mathbf{v}$ by Eq. (23). The ADMM algorithm to compute $\mathbf{v}$ is summarized in Algorithm 1.

Algorithm 1 ADMM for computing $\mathbf{v}$

Input: $\mathbf{X}$, $\mathbf{W}$, $\mathbf{G}$, $\mathbf{F}$.
Output: $\mathbf{v}$.

```
Initialize μ2 = 1, α = 1.1, v1 = v2 = 0. Compute 2(W^T W) ⊙ (X X^T) and 2diag(X F G^T W).
Compute the SVD decomposition of 2(W^T W) ⊙ (X X^T) by Eq. (20).
while not converge do
    Compute v by Eq. (23).
    Compute v1 and v2 by Eq. (16).
    Update the Lagrange multipliers by Eq. (19).
end while
```

3.2.3. Optimizing G

When optimizing $\mathbf{G}$, we obtain the following subproblem:

$$\min_{\mathbf{G}} \quad \|\mathbf{W}\mathrm{diag}(\mathbf{v})\mathbf{X} - \mathbf{G}\mathbf{F}^T\|_F^2. \tag{24}$$

Since it is an unconstraint problem, we can set the partial derivative w.r.t. $\mathbf{G}$ to zero, and obtain

$$\mathbf{G} = \mathbf{W}\mathrm{diag}(\mathbf{v})\mathbf{X}\mathbf{F}(\mathbf{F}^T\mathbf{F})^{-1} \tag{25}$$

Note that since $\mathbf{F}$ is an indicator matrix, $\mathbf{F}^T\mathbf{F}$ is a c-by-c diagonal matrix whose inverse can be computed in $O(c)$ time.

3.2.4. Optimizing F

When fixing the other variables, we obtain the following formula:

$$\begin{aligned} \min_{\mathbf{F}} \quad & \|\mathbf{W}\mathrm{diag}(\mathbf{v})\mathbf{X} - \mathbf{G}\mathbf{F}^T\|_F^2 + \sum_{m=1}^{c}\gamma_m\left(p_m - \frac{\sum_{i=1}^n F_{im}}{n}\right) \\ & + \frac{\mu_1}{2}\sum_{m=1}^{c}\left(p_m - \frac{\sum_{i=1}^n F_{im}}{n}\right)^2 \\ s.t. \quad & \mathbf{F} \in \{0,1\}^{n\times c}, \quad \sum_{m=1}^{c} F_{im} = 1. \end{aligned} \tag{26}$$

We solve $\mathbf{F}$ row by row, i.e., we optimize one row of $\mathbf{F}$ while fixing the other rows, and repeat it until convergence.

When optimizing the ith row, let $\mathbf{d}_{.i}$ denote the ith column vector of $\mathbf{W}\mathrm{diag}(\mathbf{v})\mathbf{X}$, and then we have

$$\begin{aligned} \min_{\mathbf{F}_{i.}} \quad & \|\mathbf{d}_{.i} - \mathbf{G}\mathbf{F}_{i.}^T\|_2^2 - \sum_{m=1}^{c}\frac{\gamma_m F_{im} + \mu_1 p_m F_{im}}{n} \\ & + \sum_{m=1}^{c}\frac{\mu_1 F_{im}^2 + \mu_1 F_{im}\sum_{j\neq i} F_{jm}}{2n^2}, \\ s.t. \quad & \mathbf{F}_{i.} \in \{0,1\}^c, \quad \sum_{m=1}^{c} F_{im} = 1. \end{aligned} \tag{27}$$

The constraint indicates that, in the ith row, only one element of $\mathbf{F}_{i.}$ is 1 and the others are 0s. Therefore, we can generate c candidates $\mathbf{f}_1, \ldots, \mathbf{f}_c$ where $\mathbf{f}_m \in \{0,1\}^c$ is a vector whose mth element is 1 and other elements are 0s. Then we take $\mathbf{f}_1, \ldots, \mathbf{f}_c$ into the objective function in Eq. (27), select the minimum one, and set $\mathbf{F}_{i.}$ as it.

3.2.5. Optimizing $\mathbf{p}$

When optimizing $\mathbf{p}$, we find that $\mathbf{p}$ can be decoupled into c independent subproblems and the mth one is:

$$\min_{p_m} \quad \mathcal{J} = \lambda p_m \log(p_m) + \gamma_m p_m + \frac{\mu_1}{2} p_m^2 - \mu_1 p_m a_m. \tag{28}$$

where $a_m = \frac{\sum_{i=1}^{n} F_{im}}{n} < 1$. Set its derivative w.r.t. p_m to zero, we obtain:

$$\frac{\partial \mathcal{J}}{\partial p_m} = \lambda \log(p_m) + \mu_1 p_m + \gamma_m + \lambda - \mu_1 a_m = 0. \tag{29}$$

Denote $f(p_m) = \lambda \log(p_m) + \mu_1 p_m + \gamma_m + \lambda - \mu_1 a_m$. We have $\lim_{p_m \to 0^+} f(p_m) \to -\infty$ and $f(1) = \lambda + \gamma_m + \mu_1(1 - a_m) > 0$. Moreover $f(p_m)$ is monotonically increasing in the range $(0, 1]$. Therefore, $f(p_m) = 0$ has and only has one solution in $(0, 1]$. We can solve it by a standard root finding algorithm easily.

3.2.6. Updating Lagrange multipliers

We update the Lagrange multipliers by the following formula:

$$\begin{cases} \gamma_m = \gamma_m + \mu_1 (p_m - \frac{\sum_{i=1}^{n} F_{im}}{n}), \quad 1 \le m \le c \\ \mu_1 = \beta \mu_1. \end{cases} \tag{30}$$

where $\beta > 1$ is a given parameter.

Algorithm 2 Unsupervised feature selection for balanced clustering

Input: Data matrix $\mathbf{X}$, parameters λ and τ.
Output: Feature selection vector $\mathbf{v}$.

1: Initialize $\mu_1 = 1$, $\beta = 1.1$. Compute $\mathbf{W}$ by conducting PCA on $\mathbf{X}$. Initialize $\mathbf{v} = \mathbf{1}$, and compute $\mathbf{G}$ and $\mathbf{F}$ by doing k-means on $\mathbf{W}\mathrm{diag}(\mathbf{v})\mathbf{X}$. Initialize $p_m = \frac{\sum_{i=1}^{n} F_{im}}{n}$.
2: **while** not converge **do**
3: Compute $\mathbf{W}$ by Eq. (8).
4: Compute $\mathbf{v}$ by ADMM algorithm as introduced in Algorithm 1.
5: Compute $\mathbf{G}$ by Eq. (25).
6: Compute $\mathbf{F}$ by solving Eq. (27).
7: Compute $\mathbf{p}$ by solving Eq. (29).
8: Update the Lagrange multipliers by Eq. (30).
9: **end while**

Algorithm 2 summarizes the whole ADMM algorithm.

3.3. Time and space complexity

Since we need to handle d-by-n matrix $\mathbf{X}$ and compute d-by-d matrix $\mathbf{X}\mathbf{X}^T$, the space complexity is $O(nd + d^2)$.

When computing $\mathbf{W}$, the time complexity is $O(ndk + nck + dck + nk^2 + k^3)$. When applying ADMM to compute $\mathbf{v}$, supposing that the number of iterations is l_1, the time complexity is $O(d^3 + nd^2 + cd^2 + ncd + c^2 d + l_1 d^2)$. Computing $\mathbf{G}$ costs $O(dc^2 + nc^2 + ndc)$ time. Updating $\mathbf{F}$ costs $O(nc)$ time. When updating $\mathbf{p}$, supposing the time complexity of the root finding algorithm is $O(t)$, which is independent with n and d, then it takes $O(ct)$ time to compute $\mathbf{p}$. At last, suppose the number of iterations of Algorithm 1 (Lines 2–9 in Algorithm 1) is l_2, the whole time complexity is $O(l_2(d^3 + nd^2 + l_1 d^2))$, since $c, k \ll n, d$. Although we speed up the method to some extent, we will study how to further reduce the computation complexity in the future.

Table 1
Description of the data sets.

	#instances	#features	#classes
Yale	165	1024	15
20NG	3970	1000	4
Jaffe	213	676	10
Mnist4000	4000	784	10
ORL	400	1024	40
UCI Digit	2000	216	10

4. Experiments

In this section, we compare our method with several state-of-the-art unsupervised feature selection methods on benchmark data sets.

4.1. Data sets

We conduct experiments on 9 benchmark data sets, including Yale [47], 20NG [48], Jaffe [49], Mnist4000 [50], ORL [51] and UCI Digit [52] data sets. The basic information of these data sets are summarized in Table 1.

4.2. Compared methods

To evaluate the effectiveness of our FSBC method, we compare it with the following state-of-the-art unsupervised feature selection methods:

- **AllFea**. We use all features for clustering.
- **FSASL** [30]. It learns the adaptive global and local structure in the process of feature selection.
- **SOGFS** [31]. It learns a graph with optimal structure for unsupervised feature selection.
- **LRPFS** [53]. It is a feature selection method which tries to preserve the low rank structure in the process of feature selection.
- **URAFS** [32]. This unsupervised feature selection method applies the generalized uncorrelated regression to learn an adaptive graph for feature selection.
- **NSSLFS** [29]. It is an unsupervised feature selection with sparse subspace learning.
- **FSBC_nobal**. To demonstrate the effectiveness of the balanced term $\sum_{m=1}^{c} p_m \log(p_m)$, we compare our method with **FSBC_nobal** which drops this balanced term (or equivalently speaking, set $\lambda = 0$).

4.3. Experimental setup

With the selected features, we evaluate the performance in terms of k-means clustering by Accuracy (ACC) and Normalized Mutual Information (NMI). Moreover, we also use Normalized Entropy [43,44] to evaluate the balance of the clustering results. Normalized Entropy is computed as follows:

$$NE = -\frac{1}{\log(c)} \sum_{m=1}^{c} \frac{n_m}{n} \log\left(\frac{n_m}{n}\right) \tag{31}$$

where c is the number of clusters, n is the number of instances, n_m is the number of instances in the m cluster. The larger NE is, the more balanced the clustering result is. We report the results over different number of selected features (in the range $\{20, 40, \ldots, 200\}$). Since the value of the term $\sum_{m=1}^{c} p_m \log(p_m)$ is often much smaller than other terms, we tune its parameter λ in the range $n^2 * [10^{-3}, 10^3]$ by the grid search. We tune τ in the range $[10^{-3}, 10^3]$. For other compared methods, we tune the parameters as suggested in their papers. For all methods on all data sets, the number of clusters is set to the true number of classes.

Table 2
ACC results on different data sets.

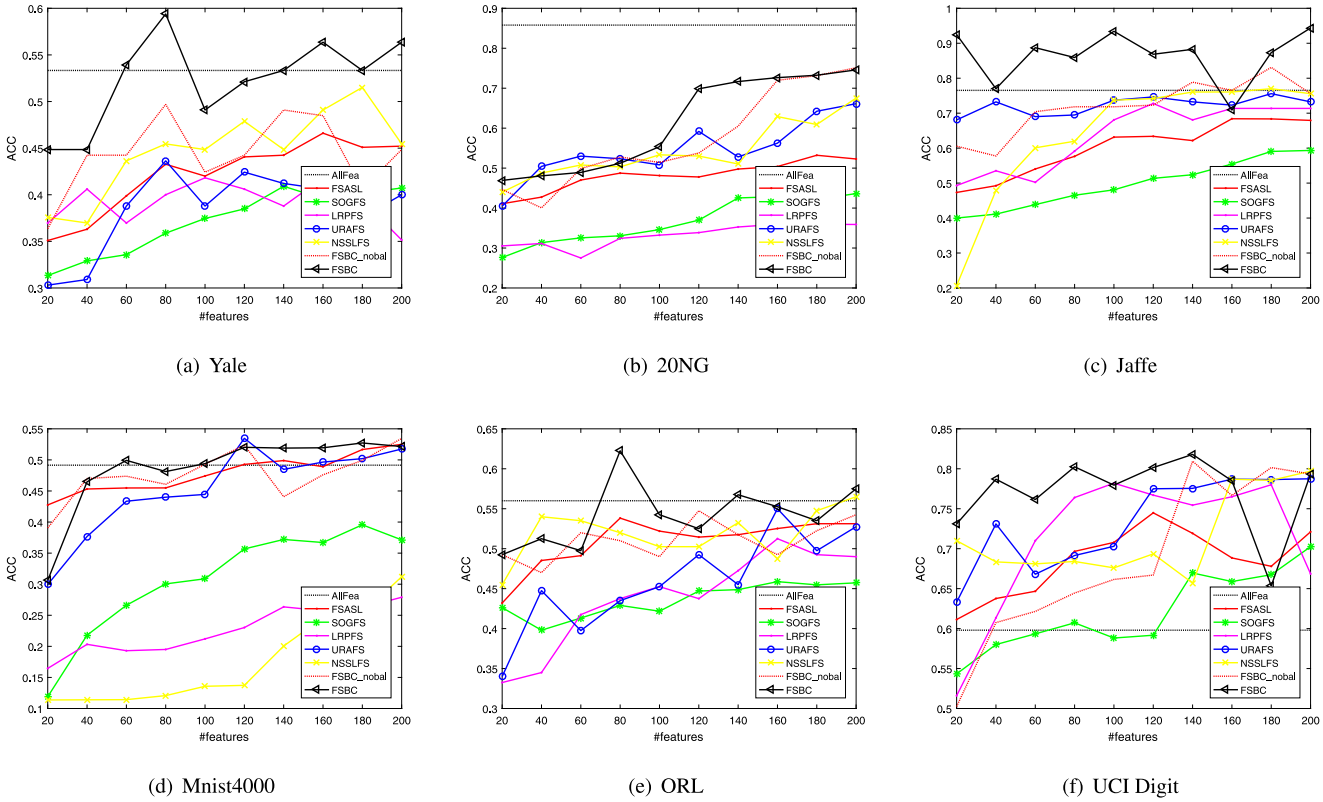
Data sets	AllFea	FSASL	SOGFS	LRPFS	URAFS	NSSLFS	FSBC_nobal	FSBC
Yale	0.5333	0.4218	0.3710	0.3921	0.3842	0.4473	0.4442	**0.5236**
20NG	0.8579	0.4812	0.3674	0.3317	0.5456	0.5425	0.5733	**0.6125**
Jaffe	0.7653	0.6016	0.4972	0.6352	0.7225	0.6432	0.7188	**0.8653**
Mnist4000	0.4915	**0.4787**	0.3075	0.2260	0.4530	0.1748	**0.4762**	**0.4853**
ORL	0.5600	0.5089	0.4355	0.4390	0.4595	0.5188	0.5105	**0.5423**
UCI Digit	0.5980	0.6852	0.6202	0.7121	0.7338	0.7154	0.6875	**0.7713**

Table 3
NMI results on different data sets.

Data sets	AllFea	FSASL	SOGFS	LRPFS	URAFS	NSSLFS	FSBC_nobal	FSBC
Yale	0.5851	0.4622	0.4133	0.4400	0.4179	0.4902	0.4756	**0.5644**
20NG	0.6605	0.2592	0.1232	0.0888	0.3080	0.3193	0.3518	**0.3813**
Jaffe	0.8473	0.6408	0.4898	0.6486	0.7261	0.6776	0.7648	**0.8632**
Mnist4000	0.4767	**0.4405**	0.2481	0.1257	0.3958	0.1020	0.4216	**0.4312**
ORL	0.7415	0.7072	0.6509	0.6412	0.6594	0.7138	0.7205	**0.7425**
UCI Digit	0.6070	0.6239	0.5852	0.6250	0.6497	0.6414	0.6181	**0.6690**

Table 4
NE results on different data sets.

Data sets	AllFea	FSASL	SOGFS	LRPFS	URAFS	NSSLFS	FSBC_nobal	FSBC
Yale	0.9734	0.9446	0.9465	0.9594	0.9369	0.9639	0.9487	**0.9771**
20NG	0.9824	0.6676	0.4600	0.3561	0.7531	**0.7856**	0.7665	**0.7967**
Jaffe	0.9700	0.9449	0.9345	0.9660	0.9809	0.9054	0.9325	**0.9916**
Mnist4000	0.9810	0.9746	0.8307	0.6534	0.9745	0.4068	0.9849	**0.9916**
ORL	0.9494	0.9500	0.9520	0.9460	0.9487	0.9547	0.9533	**0.9623**
UCI Digit	0.9531	0.9802	0.9749	**0.9904**	**0.9896**	0.9841	0.9864	**0.9932**



(a) Yale (b) 20NG (c) Jaffe

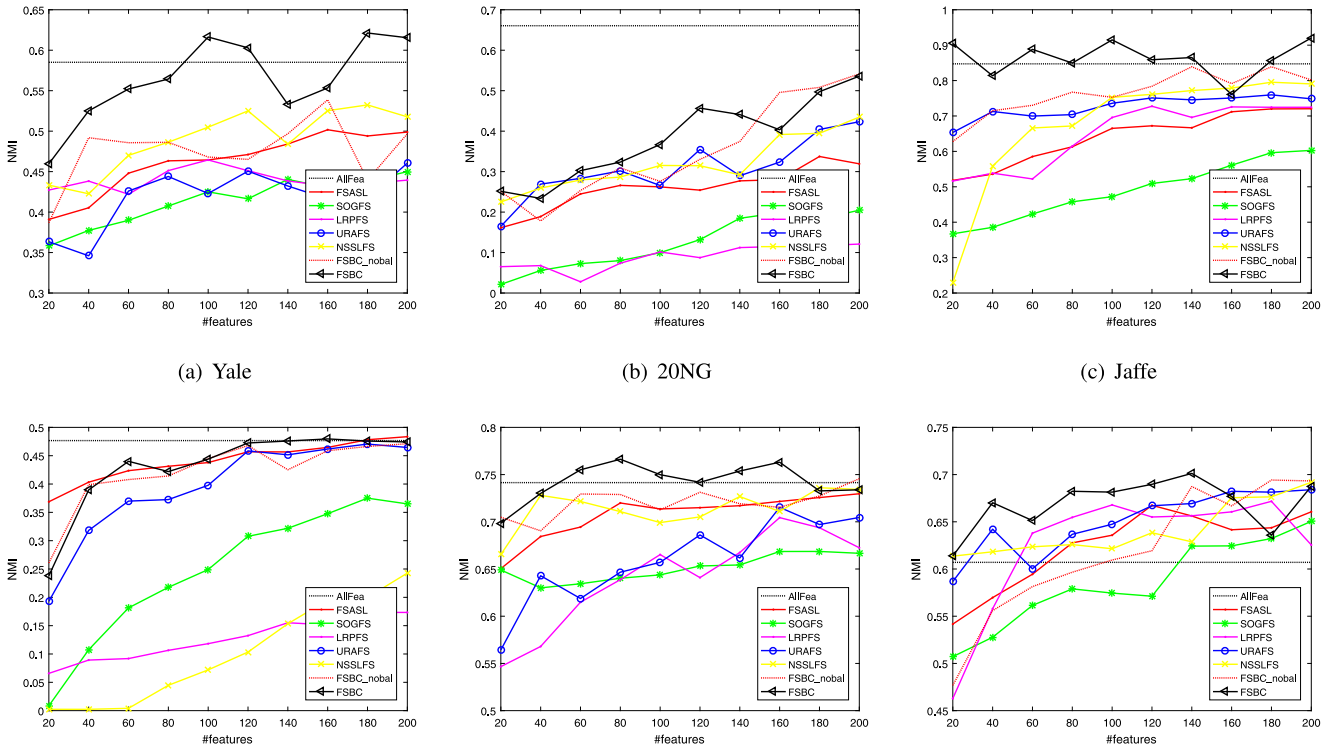
(d) Mnist4000 (e) ORL (f) UCI Digit

Fig. 1. ACC results on all data sets.

4.4. Experimental results

Since the optimal number of selected features is unknown in advance, to better evaluate the performance of unsupervised feature selection algorithms, we finally report the averaged results over different number of selected features (in the range $\{20, 40, \ldots, 200\}$). To validate the statistic significance of results, we also calculate the p-value of t-test.

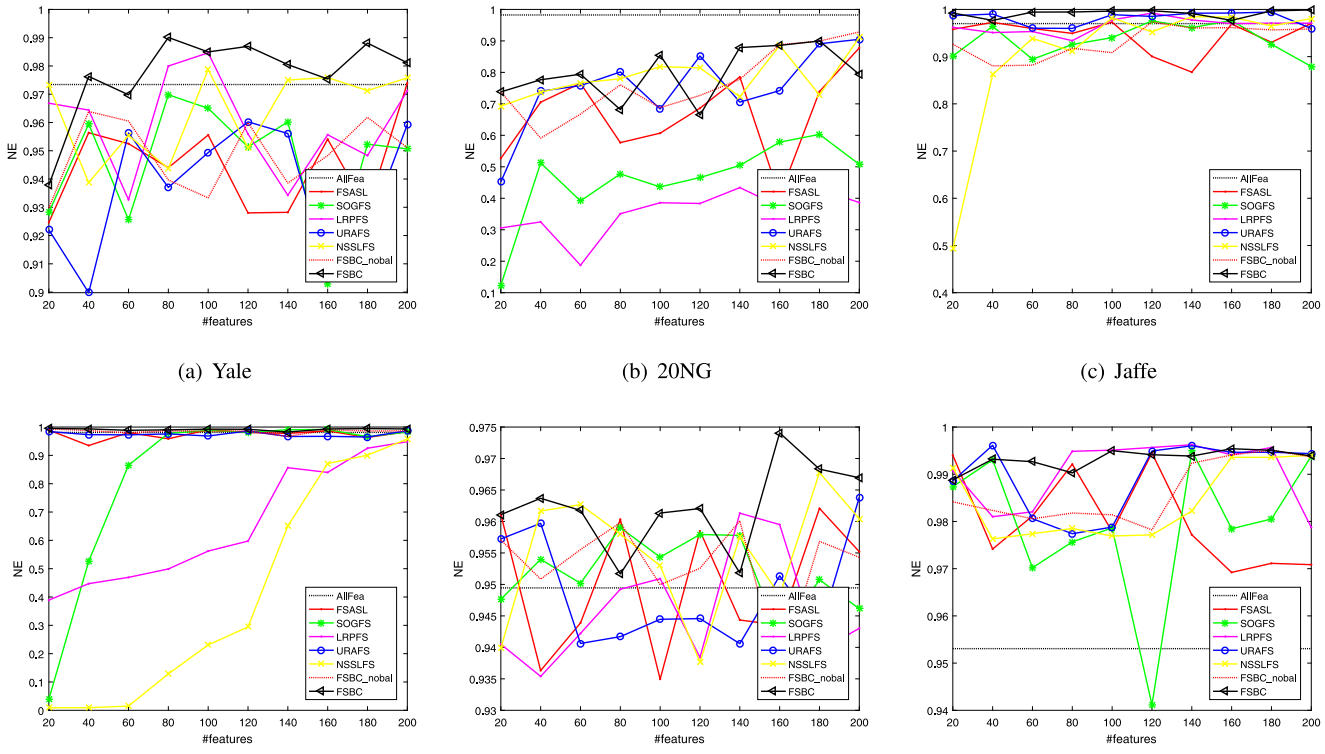
Tables 2, 3, and 4 show the results of ACC, NMI and NE on all data sets, respectively. The bold fond indicates that the difference is statistically significant, i.e., the p-value of t-test is smaller than 0.05. Note that since we aim to compare with other



(a) Yale (b) 20NG (c) Jaffe

(d) Mnist4000 (e) ORL (f) UCI Digit

Fig. 2. NMI results on all data sets.

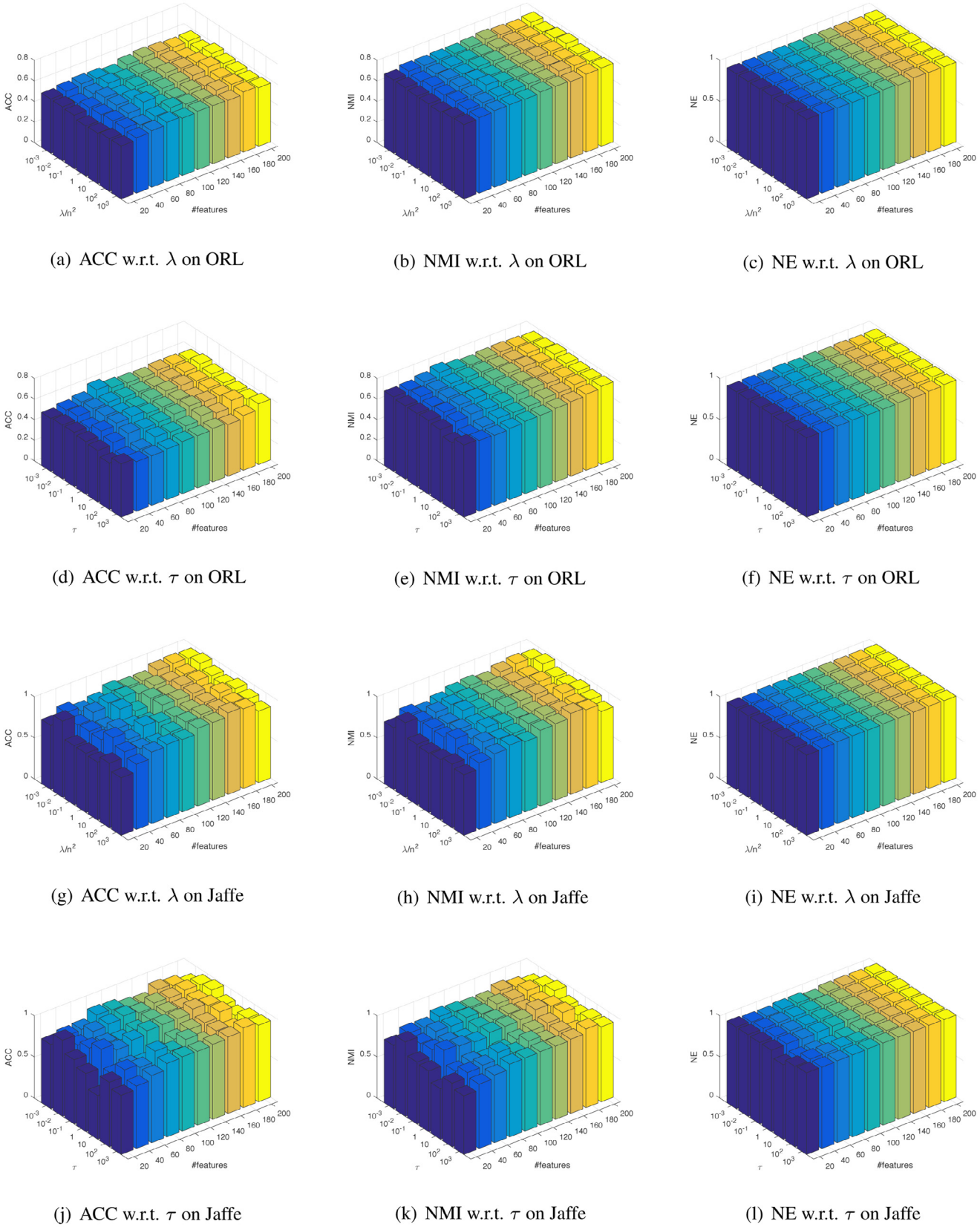


(a) Yale (b) 20NG (c) Jaffe

(d) Mnist4000 (e) ORL (f) UCI Digit

Fig. 3. NE results on all data sets.

feature selection methods, we do not calculate the p-value of AllFea. From the tables, we can see that, on most data sets, our method can outperform the state-of-the-art unsupervised feature selection methods, not only on the clustering accuracy but



(a) ACC w.r.t. λ on ORL (b) NMI w.r.t. λ on ORL (c) NE w.r.t. λ on ORL

(d) ACC w.r.t. τ on ORL (e) NMI w.r.t. τ on ORL (f) NE w.r.t. τ on ORL

(g) ACC w.r.t. λ on Jaffe (h) NMI w.r.t. λ on Jaffe (i) NE w.r.t. λ on Jaffe

(j) ACC w.r.t. τ on Jaffe (k) NMI w.r.t. τ on Jaffe (l) NE w.r.t. τ on Jaffe

Fig. 4. ACC, NMI and NE w.r.t λ and τ on ORL and Jaffe data sets.

also the balance. This well demonstrates the effectiveness of our FSBC. The reason why our method outperforms others may be in two folds. Firstly, our method considers the balance property of data. Note that, on most data sets, our method significantly

outperforms the FSBC_nobal, which drops the balanced term. It demonstrates that the balanced regularized term can indeed improve the performance (both on clustering accuracy and balance) of our method. Secondly, different from other feature selection methods which use $\ell_{2,1}$-norm to approximate the $\ell_{2,0}$-norm, we directly apply $\ell_{2,0}$-norm to select the exact top-k features without any approximation, which is more desirable in feature selection task. Therefore, our method performs better in both the clustering quality and revealing balance structure.

The details of ACC, NMI and NE results on each data set with different number of features are show in Figs. 1, 2, and 3. In these figures, the black line represents our method, and the black horizontal dotted line represents the result of AllFea. We can see that, on most data sets, our method can outperform the AllFea at most time. It demonstrates that our method can not only largely reduce the number of features used for clustering, but also often improve the clustering performance. It can also be found that our method outperforms the state-of-the-art feature selection methods on most data sets at most time. The red dotted line represents the result of FSBC_nobal, which is often below our method. It also demonstrates the necessity of the balanced regularized term.

4.5. Parameter study

We explore the affect of the parameters on clustering performance by tuning parameters λ in $n^2 * [10^{-3}, 10^3]$ and τ in $[10^{-3}, 10^3]$. Fig. 4 shows the results on ORL and Jaffe data sets, and the results on other data sets are similar. The results show that the performance of our method is stable across a wide range of the parameters, thus we can choose the parameters easily.

5. Conclusion

In this paper, we proposed a novel unsupervised feature selection method for balanced clustering. Different from the conventional feature selection methods which selected the discriminative features, we focused on the features which can reveal the balanced structure of the data. We proposed a balance regularization term and applied it to k-means, leading to a balanced k-means clustering. In this balanced k-means clustering framework, we selected the informative features which can make the clustering results as balanced as possible. We integrated the balanced k-means and feature selection into a unified framework and provided an effective ADMM algorithm to jointly do clustering and select features. At last, we conduct extensive experiments on benchmark data sets, and the experimental results show the superiority of our method.

The proposed FSBC method can be very useful for the scenarios which need to capture the balanced structure of data. Although we have already speeded up the method to some extent, this method still suffers from the high time and space complexity. In the future, we will consider this scalable issue and try to further reduce the time and space complexity of our method.

CRediT authorship contribution statement

Peng Zhou: Conceptualization, Methodology, Software, Writing - original draft. **Jiangyong Chen:** Methodology, Software. **Mingyu Fan:** Software, Writing - review & editing. **Liang Du:** Writing - review & editing, Funding acquisition. **Yi-Dong Shen:** Supervision, Funding acquisition. **Xuejun Li:** Funding acquisition.

Acknowledgments

This work is supported by the National Natural Science Fund of China grants 61806003, 61772373, 61976129, and 61972001, and the Key Natural Science Project of Anhui Provincial Education Department, China KJ2018A0010.

References

[1] T. Althoff, A. Ulges, A. Dengel, Balanced clustering for content-based image browsing, Ser. Gesellschaft fur Inf. 1 (2011) 27–30.
[2] Z. Du, Y. Liu, D. Qian, An energy-efficient balanced clustering algorithm for wireless sensor networks, in: 2009 5th International Conference on Wireless Communications, Networking and Mobile Computing, IEEE, 2009, pp. 1–4.
[3] S. Zhong, J. Ghosh, Model-based clustering with soft balancing, in: ICDM, 2003, pp. 459–466.
[4] R. Tibshirani, Regression shrinkage and selection via the lasso, J. R. Stat. Soc. Ser. B Stat. Methodol. (1996) 267–288.
[5] J.G. Dy, C.E. Brodley, Feature selection for unsupervised learning, J. Mach. Learn. Res. 5 (Aug) (2004) 845–889.
[6] F. Nie, H. Huang, X. Cai, C.H. Ding, Efficient and robust feature selection via joint $\ell_{2,1}$-norms minimization, in: Advances in Neural Information Processing Systems, 2010, pp. 1813–1821.
[7] Z. Xu, I. King, M.R.-T. Lyu, R. Jin, Discriminative semi-supervised feature selection via manifold regularization, IEEE Trans. Neural Netw. 21 (7) (2010) 1033–1047.
[8] Z. Zhang, F. Li, M. Zhao, L. Zhang, S. Yan, Robust neighborhood preserving projection by nuclear/L2,1-norm regularization for image feature extraction, IEEE Trans. Image Process. 26 (4) (2017) 1607–1622.
[9] X. Zhu, S. Zhang, R. Hu, Y. Zhu, J. Song, Local and global structure preservation for robust unsupervised spectral feature selection, IEEE Trans. Knowl. Data Eng. 30 (3) (2018) 517–529.
[10] M. Luo, F. Nie, X. Chang, Y. Yang, A.G. Hauptmann, Q. Zheng, Adaptive unsupervised feature selection with structure regularization, IEEE Trans. Neural Netw. Learn. Syst. 29 (4) (2018) 944–956.
[11] L.M.Q. Abualigah, Feature Selection and Enhanced Krill Herd Algorithm for Text Document Clustering, Springer, 2019.
[12] Y. Yang, H.T. Shen, Z. Ma, Z. Huang, X. Zhou, L21-norm regularized discriminative feature selection for unsupervised learning, in: IJCAI, 2011.
[13] P. Zhu, Q. Xu, Q. Hu, C. Zhang, Co-regularized unsupervised feature selection, Neurocomputing 275 (2018) 2855–2863.
[14] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, et al., Distributed optimization and statistical learning via the alternating direction method of multipliers, Found. Trends Mach. Learn. 3 (1) (2011) 1–122.
[15] H. Tao, C. Hou, F. Nie, Y. Jiao, D. Yi, Effective discriminative feature selection with nontrivial solution, IEEE Trans. Neural Netw. Learn. Syst. 27 (4) (2015) 796–808.
[16] Q. Gu, Z. Li, J. Han, Generalized fisher score for feature selection, in: Conference on Uncertainty in Artificial Intelligence, 2011, pp. 266–273.
[17] M. Fan, X. Chang, X. Zhang, D. Wang, L. Du, Top-k supervise feature selection via ADMM for integer programming, in: IJCAI, 2017, pp. 1646–1653.
[18] Y. Han, Y. Yang, Y. Yan, Z. Ma, N. Sebe, X. Zhou, Semisupervised feature selection via spline regression for video semantic recognition, IEEE Trans. Neural Netw. Learn. Syst. 26 (2) (2014) 252–264.
[19] X. Chang, F. Nie, Y. Yang, H. Huang, A convex formulation for semi-supervised multi-label feature selection, in: Twenty-Eighth AAAI Conference on Artificial Intelligence, 2014.
[20] X. Chang, Y. Yang, Semisupervised feature analysis by mining correlations among multiple tasks, IEEE Trans. Neural Netw. Learn. Syst. 28 (10) (2016) 2294–2305.
[21] M. Luo, X. Chang, L. Nie, Y. Yang, A.G. Hauptmann, Q. Zheng, An adaptive semisupervised feature analysis for video semantic recognition, IEEE Trans. Cybern. 48 (2) (2017) 648–660.
[22] P. Zhu, W. Zuo, L. Zhang, Q. Hu, S.C. Shiu, Unsupervised feature selection by regularized self-representation, Pattern Recognit. 48 (2) (2015) 438–446.
[23] C. Hou, F. Nie, X. Li, D. Yi, Y. Wu, Joint embedding learning and sparse regression: A framework for unsupervised feature selection, IEEE Trans. Cybern. 44 (6) (2013) 793–804.
[24] S. Wang, J. Tang, H. Liu, Embedded unsupervised feature selection, in: Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015.
[25] L. Shi, L. Du, Y.-D. Shen, Robust spectral learning for unsupervised feature selection, in: 2014 IEEE International Conference on Data Mining, IEEE, 2014, pp. 977–982.
[26] L.M. Abualigah, A.T. Khader, Unsupervised text feature selection technique based on hybrid particle swarm optimization algorithm with genetic operators for the text clustering, J. Supercomput. 73 (11) (2017) 4773–4795.
[27] L.M. Abualigah, A.T. Khader, E.S. Hanandeh, A new feature selection method to improve the document clustering using particle swarm optimization algorithm, J. Comput. Sci. 25 (2018) 456–466.
[28] M. Fan, X. Chang, D. Tao, Structure regularized unsupervised discriminant feature analysis, in: Thirty-First AAAI Conference on Artificial Intelligence, 2017.
[29] W. Zheng, H. Yan, J. Yang, Robust unsupervised feature selection by nonnegative sparse subspace learning, Neurocomputing 334 (2019) 156–171.

[30] L. Du, Y.-D. Shen, Unsupervised feature selection with adaptive structure learning, in: SIGKDD, ACM, 2015, pp. 209–218.
[31] F. Nie, W. Zhu, X. Li, et al., Unsupervised feature selection with structured graph optimization, in: AAAI, 2016, pp. 1302–1308.
[32] X. Li, H. Zhang, R. Zhang, Y. Liu, F. Nie, Generalized uncorrelated regression with adaptive graph for unsupervised feature selection, IEEE Trans. Neural Netw. Learn. Syst. (2018).
[33] A.Y. Ng, M.I. Jordan, Y. Weiss, On spectral clustering: Analysis and an algorithm, in: NIPS, 2001, pp. 849–856.
[34] L.M. Abualigah, A.T. Khader, E.S. Hanandeh, A.H. Gandomi, A novel hybridization strategy for krill herd algorithm applied to clustering techniques, Appl. Soft Comput. 60 (2017) 423–435.
[35] L.M. Abualigah, A.T. Khader, E.S. Hanandeh, Hybrid clustering analysis using improved krill herd algorithm, Appl. Intell. 48 (11) (2018) 4047–4071.
[36] L.M. Abualigah, A.T. Khader, E.S. Hanandeh, A combination of objective functions and hybrid krill herd algorithm for text document clustering analysis, Eng. Appl. Artif. Intell. 73 (2018) 111–125.
[37] P. Zhou, F. Ye, L. Du, Unsupervised robust multiple kernel learning via extracting local and global noises, IEEE Access 7 (2019) 34451–34461.
[38] P. Bradley, K. Bennett, A. Demiriz, Constrained k-means Clustering, Microsoft Research, Redmond, 2000, p. 20.
[39] M.I. Malinen, P. Fränti, Balanced k-means for clustering, in: Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition, SPR, and Structural and Syntactic Pattern Recognition, SSPR, Springer, 2014, pp. 32–41.
[40] L.R. Costa, D. Aloise, N. Mladenović, Less is more: basic variable neighborhood search heuristic for balanced minimum sum-of-squares clustering, Inform. Sci. 415 (2017) 247–253.
[41] A. Banerjee, J. Ghosh, On scaling up balanced clustering algorithms, in: Proceedings of the 2002 SIAM International Conference on Data Mining, SIAM, 2002, pp. 333–349.
[42] A. Banerjee, J. Ghosh, Frequency-sensitive competitive learning for scalable balanced clustering on high-dimensional hyperspheres, IEEE Trans. Neural Netw. 15 (3) (2004) 702–719.
[43] H. Liu, J. Han, F. Nie, X. Li, Balanced clustering with least square regression, in: Thirty-First AAAI Conference on Artificial Intelligence, 2017.
[44] Z. Li, F. Nie, X. Chang, Z. Ma, Y. Yang, Balanced clustering via exclusive lasso: A pragmatic approach, in: Thirty-Second AAAI Conference on Artificial Intelligence, 2018.
[45] I. Jolliffe, Principal Component Analysis, Springer, 2011.
[46] B. Wu, B. Ghanem, Lp-box ADMM: A versatile framework for integer programming, IEEE Trans. Pattern Anal. Mach. Intell. (2018).
[47] D. Cai, X. He, Y. Hu, J. Han, T. Huang, Learning a spatially smooth subspace for face recognition, in: 2007 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2007, pp. 1–7.
[48] D. Cai, X. He, W.V. Zhang, J. Han, Regularized locality preserving indexing via spectral regression, in: Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management, ACM, 2007, pp. 741–750.
[49] M.J. Lyons, J. Budynek, S. Akamatsu, Automatic classification of single facial images, IEEE Trans. Pattern Anal. Mach. Intell. 21 (12) (1999) 1357–1362.
[50] D. Cai, X. He, J. Han, T.S. Huang, Graph regularized nonnegative matrix factorization for data representation, IEEE Trans. Pattern Anal. Mach. Intell. 33 (8) (2010) 1548–1560.
[51] F.S. Samaria, A.C. Harter, Parameterisation of a stochastic model for human face identification, in: Proceedings of 1994 IEEE Workshop on Applications of Computer Vision, IEEE, 1994, pp. 138–142.
[52] P. Zhou, Y.-D. Shen, L. Du, F. Ye, X. Li, Incremental multi-view spectral clustering, Knowl.-Based Syst. 174 (2019) 73–86.
[53] W. Zheng, C. Xu, J. Yang, J. Gao, F. Zhu, Low-rank structure preserving for unsupervised feature selection, Neurocomputing 314 (2018) 360–370.