Joint Active Learning with Feature Selection via CUR Matrix Decomposition

Changsheng Li, Xiangfeng Wang, Weishan Dong, Junchi Yan, Qingshan Liu, *Senior Member, IEEE,* and Hongyuan Zha

Abstract—This paper focuses on the problem of simultaneous sample and feature selection for machine learning in a fully unsupervised setting. Though most existing works tackle these two problems separately that derives two well-studied sub-areas namely active learning and feature selection, a unified approach is inspirational since they are often interleaved with each other. Noisy and high-dimensional features will bring adverse effect on sample selection, while 'good' samples will be beneficial to feature selection. We present a framework to jointly conduct active learning and feature selection based on the CUR matrix decomposition. From the data reconstruction perspective, both the selected samples and features can best approximate the original dataset respectively, such that the selected samples characterized by the selected features are very representative. Additionally our method is one-shot without iteratively selecting samples for progressive labeling. Thus our model is especially suitable when the initial labeled samples are scarce or totally absent, which existing works hardly address particularly for simultaneous feature selection. To alleviate the NP-hardness of the raw problem, the proposed formulation involves a convex but non-smooth optimization problem. We solve it efficiently by an iterative algorithm, and prove its global convergence. Experiments on publicly available datasets validate that our method is promising compared with the state-of-the-arts.

Index Terms—Active learning, feature selection, matrix factorization

1 Introduction

In many real-life machine learning tasks, unlabeled data are often easily available whereas labeled data are scarce. In order to build powerful predictive models, one usually requires domain experts to manually annotate samples, but this is an expensive and time-consuming procedure. Active learning [1] provides a means to alleviate this problem by carefully selecting samples to be labeled by experts. Typically, the active learning algorithms prefer to query those unlabeled samples which can improve the prediction performance the most if they were labeled and used as training data. In this way, the active learner aims to pick out as few samples as possible to label for minimizing the total annotating cost, while an accurate supervised learning model can be built based on these labeled data.

In the past decade, lots of active learning algorithms have been proposed [2], [3], [4], [5], [6], and have been successfully applied to a variety of problems in computer vision [7], [8], [9], [10], [11], [12]. Generally speaking, there are two main group methods for selecting unlabeled samples to label [13]: One is to select the most informative samples, such as uncertainty sampling [14], [15], query by committee [1], and empirical risk minimization [16]. These algorithms are implemented iteratively, where a model is learned with the existing labeled data and new samples are chosen to be labeled based on the learned model. Since training model usually needs a large number of labeled data to avoid the samples bias, the methods above should be used after sufficient labeled samples are collected [17]. The other group aims at querying the most representative samples from a perspective of data reconstruction [17], [18], [19], [20], [21]. Different from the first group, methods in this group are one-shot and non-iterative for selecting samples. Such active learning methods are usually applied when there is no initial labeled data.

Although active learning has been well studied for years, it still has some issues in many real-world scenarios. For example, the sample is often characterized by high-dimensional features, and some of features are often noisy or irrelevant. These noisy or irrelevant features bring adverse influence on selecting informative or representative samples. Moreover, after querying samples, some supervised learning models, such as decision tree, are often trained based on these labeled data for various applications. However, high-dimensional features significantly increase the time and space requirements for model training. Meanwhile, when only limited labeled samples are available, it is difficult to guarantee reliable model parameter estimates in high-dimensional feature space. One may state that, if we apply some state-of-the-art feature selection techniques, such as SPEC [22], $Q-\alpha$ [23], to learn a low-dimensional representation before active learning, these problems might be solved. Of course, this should be helpful for active learning to some extent, while common feature selection techniques and active learning algorithms are independent in designing, directly combining them usually cannot guarantee to obtain the optimal results. Therefore, it will benefit from devising principled model and algorithm for incorporating active learning and feature selection in a unified fash-

- *C. Li and W. Dong are with IBM Research-China, Beijing, China. E-mail: {lcsheng,dongweis}@cn.ibm.com*
- *X. Wang and J. Yan are with East China Normal University, Shanghai, China. E-mail: {xfwang,jcyan}@sei.ecnu.edu.cn*
- *Q. Liu is with Nanjing University of Information Science and Technology. E-mail: qsliu@nuist.edu.cn*
- *H. Zha is with Georgia Institute of Technology, Atlanta, USA. E-mail: zha@cc.gatech.edu*

ion. Recently, Joshi and Xu [24] presented an active learning method with integrated feature selection based on linear kernel SVMs and GainRatio. Raghavan et al. [25] intended to use human feedback on both features and samples for active learning. Kong et al. [26] proposed a dual feature selection and sample selection method in the context of graph classification. Bilgic [27] proposed a dynamic dimensionality reduction algorithm that determined the appropriate number of dimensions for each active learning iteration. Since all of the above three algorithms are implemented iteratively, and need to train models for querying in each iteration, they are suitable to work in the scenarios of the first group active learning methods. Different from them, we focus on studying the problem of the second active learning group, i.e., in the case when no initial labeled samples are available, by jointly learning important features and samples. This is an unsupervised learning problem, which is much harder due to the absence of labels that would guide the search for relevant information.

In this paper, we present a unified view of (sampled based) Active Learning and Feature Selection, called ALFS, which is inspired by the approximation method for CUR matrix decomposition.

The main contributions of this paper are:

i) To our knowledge, this is the first work for presenting a unified view for one-shot active learning and feature selection, which is important for real-world applications, since it dispenses with any label effort unlike those progressive interactive labeling active learning methods.
ii) This work is the first one to formulate and build the natural connection between CUR decomposition and simultaneous sample and feature selection.
iii) We devise a novel model and convex optimization algorithm to solve the one-shot sample and feature learning problem.
iv) The convergence of the proposed iterative algorithm is theoretically proved, and extensive empirical results demonstrate the advantages of our approach.

The rest of this paper is organized as follows: we propose a unified framework to conduct active learning and feature selection simultaneously in section 2. Section 3 reviews related work on the second group active learning algorithms. The experimental results are reported in Section 4. Section 5 presents concluding remarks and future work.

Notations. In this paper, matrices are written as boldface uppercase letters and vectors are written as boldface lowercase letters. Given a matrix $\mathbf{P}$, we denote its (i,j)-th entry, i-th row, j-th column as $\mathbf{P}_{ij}$, $\mathbf{p}^i$, $\mathbf{p}_j$, respectively. The only used vector norm is the l_2 norm, denoted by $\|\cdot\|_2$. A variety of norms on matrices will be used. The l_1, $l_{2,1}$, l_∞ norms of a matrix are defined by $\|\mathbf{P}\|_1 = \sum_{i,j}|\mathbf{P}_{ij}|$, $\|\mathbf{P}\|_{2,1} = \sum_{i=1}^m \sqrt{\sum_{j=1}^n \mathbf{P}_{ij}^2} = \sum_{i=1}^m \|\mathbf{p}^i\|_2$, and $\|\mathbf{P}\|_\infty = \max_{i,j}|\mathbf{P}_{ij}|$, respectively. The quasi-norm $l_{2,0}$ norm of a matrix $\mathbf{P}$ is defined as the number of the nonzero rows of $\mathbf{P}$, denoted by $\|\mathbf{P}\|_{2,0}$. The Frobenius norm is denoted by $\|\mathbf{P}\|_F$. The Euclidean inner product between two matrices is $\langle \mathbf{P}, \mathbf{Q}\rangle = tr(\mathbf{P}^T\mathbf{Q})$, where $\mathbf{P}^T$ is the transpose of the matrix $\mathbf{P}$ and $tr(\cdot)$ is the trace of a matrix. The rank of a matrix is denoted by $rank(\cdot)$.

2 PROPOSED METHOD

Given an unlabeled dataset $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^{d\times n}$, our goal is to pick out m $(m < n)$ samples for labeling by user, and simultaneously select r $(r < d)$ features as the new feature representation, such that the potential performance is maximized when the model is trained based on the selected m labeled samples under the new representation. This is a more challenging problem than traditional representativeness based active learning problems, because selecting m samples to best approximate $\mathbf{X}$ often leads to an NP-hard problem [18], and finding r features as the most representative feature subset is also often NP-hard [28].

2.1 Active Learning and Feature Selection via Matrix Decomposition

Inspired by the CUR matrix decomposition [29], [30], [31], [32], we propose a unified framework to find the most representative samples and features. To make this paper self-contained, we first introduce CUR matrix factorization.

Definition 2.1. Given $\mathbf{X} \in \mathbb{R}^{d\times n}$ of rank $\rho = rank(\mathbf{X})$, rank parameter $k < \rho$, and accuracy parameter $0 < \varepsilon < 1$, the CUR factorization for $\mathbf{X}$ aims to find $\mathbf{C} \in \mathbb{R}^{d\times m}$ with m columns from $\mathbf{X}$, $\mathbf{R} \in \mathbb{R}^{r\times n}$ with r rows of $\mathbf{X}$, and $\mathbf{U} \in \mathbb{R}^{m\times r}$, with m, r, and $rank(\mathbf{U})$ being as small as possible, such that $\mathbf{X}$ is reconstructed within relative-error:

$$\|\mathbf{X} - \mathbf{CUR}\|_F^2 \le (1+\varepsilon)\|\mathbf{X} - \mathbf{X}_k\|_F^2, \qquad (1)$$

where $\mathbf{X}_k = \mathbf{U}_k\Sigma_k\mathbf{V}_k^T \in \mathbb{R}^{d\times n}$ is the best rank k matrix obtained via the SVD of $\mathbf{X}$.

From an algorithmic perspective, the matrices $\mathbf{C}$, $\mathbf{U}$, and $\mathbf{R}$ can be obtained by minimizing the approximation error $\|\mathbf{X} - \mathbf{CUR}\|_F^2$. Here we make a key observation that the above definition is closely related to the problem of simultaneous sample and feature selection, though to our surprise, existing works hardly point out or explore this connection to solve the active learning problem: on one hand, $\mathbf{UR}$ can be regarded as a reconstruction coefficient matrix, and $\mathbf{C}$ denotes the selected m samples, thus minimizing $\|\mathbf{X} - \mathbf{CUR}\|_F^2$ means that the total reconstruction error is minimized, which can make the data points listed in $\mathbf{C}$ be the most representative. The reconstruction coefficients $\mathbf{UR}$ are related to an r-dimensional feature subset of the dataset. Actually, the reconstruction coefficients of each reconstructed data point $\mathbf{x}_i$ are formed by a linear combination of its r features. On the other hand, $\mathbf{CU}$ can be also regarded as a reconstruction coefficient matrix, and $\mathbf{R}$ is the new low-dimensional representation of $\mathbf{X}$, so minimizing $\|\mathbf{X} - \mathbf{CUR}\|_F^2$ also indicates that the selected r features can represent the whole dataset most precisely. The construction of the coefficient matrix $\mathbf{CU}$ depends on a sample subset of $\mathbf{X}$. Clearly, active learning and feature selection can be conducted simultaneously in such a joint framework via CUR factorization.

Despite the above connection from CUR decomposition to feature selection and active learning, the original CUR formulation and its existing solvers can not be directly applied to solve the simultaneous feature and sample selection task due to the under-determination of a general CUR

model. In the context of active sample/feature learning, this paper proposes a tailored objective function rooted from CUR decomposition, while being more informative by adding regularization terms to incorporate prior knowledge. Moreover, unlike most existing CUR solvers being randomized or heuristic algorithms [29], [30], we utilize the structured sparsity-inducing norms to relax the objective from a non-convex optimization problem to a convex one, which allows for devising an efficient variant of the alternating direction method of multipliers (ADMM) [33], [34].

2.2 A Convex Formulation

Let $\mathbf{p} = (p_1, \ldots, p_n)^T \in \{0,1\}^n$ and $\mathbf{q} = (q_1, \ldots, q_d)^T \in \{0,1\}^d$ denote two indicator variables to represent whether a sample and a feature is selected or not, respectively. $p_i = 1$ (or 0) indicates that the i-th sample is selected (or not), and $q_i = 1$ (or 0) means that the i-th feature is selected (or not). Minimizing $\|\mathbf{X} - \mathbf{CUR}\|_F^2$ can be re-written as:

$$\begin{aligned} \min_{\mathbf{p},\mathbf{q},\widehat{\mathbf{U}}\in\mathbb{R}^{n\times d}} & \|\mathbf{X} - \mathbf{X}\mathrm{diag}(\mathbf{p})\widehat{\mathbf{U}}\mathrm{diag}(\mathbf{q})\mathbf{X}\|_F^2 \\ s.t.\ & \mathbf{1}_n^T\mathbf{p} = m, \mathbf{p} \in \{0,1\}^n, \\ & \mathbf{1}_d^T\mathbf{q} = r, \mathbf{q} \in \{0,1\}^r, \end{aligned} \tag{2}$$

where $\mathrm{diag}(\mathbf{p})$ is a diagonal matrix with its diagonal elements being $\mathbf{p}$, and $\mathbf{1}_n$ is an n-dimensional vector with all components being 1. $\mathbf{X}\mathrm{diag}(\mathbf{p})$ in (2) aims to make m columns of $\mathbf{X}$ unchanged, and reset the rest $n - m$ columns to zero vectors. $\mathrm{diag}(\mathbf{q})\mathbf{X}$ tends to keep r rows of $\mathbf{X}$ unchanged, and reset the rest $(d-r)$ rows to zero vectors.

Using the matrix $l_{2,0}$ norm, we formulate the problem in (2) as:

$$\begin{aligned} \min_{\mathbf{W}\in\mathbb{R}^{n\times d}} & \|\mathbf{X} - \mathbf{XWX}\|_F^2 \\ s.t.\ & \|\mathbf{W}\|_{2,0} = m, \|\mathbf{W}^T\|_{2,0} = r, \end{aligned} \tag{3}$$

where $\mathbf{W} = \mathrm{diag}(\mathbf{p})\widehat{\mathbf{U}}\mathrm{diag}(\mathbf{q}) \in \mathbb{R}^{n\times d}$.

Based on (3), we propose to optimize the following objective function:

$$\min_{\mathbf{W}\in\mathbb{R}^{n\times d}} \|\mathbf{X} - \mathbf{XWX}\|_F^2 + \alpha\|\mathbf{W}\|_{2,0} + \beta\|\mathbf{W}^T\|_{2,0}, \tag{4}$$

where $\alpha \geq 0$ and $\beta \geq 0$ are two regularization parameters.

However, (4) is still an NP-hard problem due to the matrix $l_{2,0}$ norm. Fortunately, there exists theoretical progress that $\|\mathbf{W}\|_{2,1}$ is the minimum convex hull of $\|\mathbf{W}\|_{2,0}$ [17]. The result of minimizing $\|\mathbf{W}\|_{2,1}$ is the same as that of minimizing $\|\mathbf{W}\|_{2,0}$, as long as $\mathbf{W}$ is row-sparse enough. Therefore, (4) can be relaxed to the following convex optimization problem:

$$\min_{\mathbf{W}\in\mathbb{R}^{n\times d}} \|\mathbf{X} - \mathbf{XWX}\|_F^2 + \alpha\|\mathbf{W}\|_{2,1} + \beta\|\mathbf{W}^T\|_{2,1}. \tag{5}$$

2.3 Local Linear Reconstruction

In the new objective function (5), we can see that each data point is reconstructed by a linear combination of all the selected points (when the i-th row of the reconstruction coefficient matrix $\mathbf{WX}$ in (5) is not a zero vector, $\mathbf{x}_i$ is chosen as one of the most representative samples. Otherwise, $\mathbf{x}_i$ is not selected). However, it is more reasonable to suppose that a data point can be mainly recovered from its neighbors [20], [21]. Intuitively, if the distance between the reconstructed point and the selected point is large, the contribution of the selected point should be small to the reconstruction of the target point, and thus the reconstruction coefficient should be penalized. In light of this point, we incorporate a regularization term into (5) as:

$$\begin{aligned} \min_{\mathbf{W}\in\mathbb{R}^{n\times d}} & \|\mathbf{X} - \mathbf{XWX}\|_F^2 + \alpha\|\mathbf{W}\|_{2,1} + \beta\|\mathbf{W}^T\|_{2,1} \\ & + \lambda\|\mathbf{T} \odot (\mathbf{WX})\|_1, \end{aligned} \tag{6}$$

where $\lambda \geq 0$ is a regularization parameter, and $\odot$ denotes the element-wise multiplication of two matrices. $\mathbf{T}$ is a weight matrix, where $\mathbf{T}_{ij}$ encodes the distance between the i-th and j-th samples. From the data reconstruction perspective, if two unit vectors have the same or opposite directions, their distance should be minimal, since either vector can be fully recovered by the other one; on the contrary, if the two vectors are orthogonal, their distance should be maximal, because they have little contribution to each other's reconstruction. Therefore, we use the absolute value of the cosine function of the angle between two feature vectors to measure their similarity, and define the inverse of the absolute value as their distance:

$$\mathbf{T}_{ij} = \frac{1}{|\cos\theta_{ij}|}, \tag{7}$$

where θ_{ij} denotes the angle between $\mathbf{x}_i$ and $\mathbf{x}_j$[1].

After obtaining the optimal $\mathbf{W}$ in (6), we can sort all the samples by the l_2 norm of the rows of $\mathbf{W}$ in descending order, and select the top m samples as the representative ones. Similarly, we rank all the features by the l_2 norm of the columns of $\mathbf{W}$ in descending order, and choose the top r features to represent the samples.

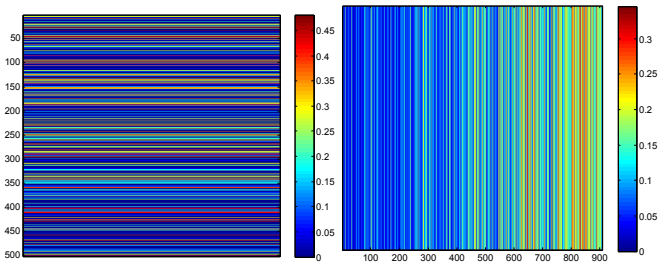
We take the FG-NET dataset[2] as an example to illustrate the effectiveness of the $l_{2,1}$ norm constraint on $\mathbf{W}$ and $\mathbf{W}^T$ in (6). Fig. 1 (a) and (b) are the visualizations of $l_{2,1}$ norm of $\mathbf{W}$ and $\mathbf{W}^T$, respectively. Many rows and columns in $\mathbf{W}$ become sparse by adding the $l_{2,1}$ norm constraints on $\mathbf{W}$ and $\mathbf{W}^T$, which means that $\mathbf{W}$ can conduct sample selection and feature selection simultaneously.

2.4 Optimization Algorithm

Although the problem (6) is convex, it is not easy to be solved by sub-gradient type methods since different structured non-smooth terms are involved. In this section, we employ the alternating direction method of multipliers (ADMM) [33] to solve (6). Theoretical results will be given then including the global convergence and iteration complexity.

1. When $\cos\theta_{ij} = 0$, we can regularize $\mathbf{T}_{ij}$ as $\mathbf{T}_{ij} = \frac{1}{|\cos\theta_{ij}|+\varsigma}$, where ς is a very small positive constant.

2. The dataset is available at http://sting.cycollege.ac.cy/alanitis/fgnetaging/index.htm.



(a) $\|\mathbf{W}\|_{2,1}$ (b) $\|\mathbf{W}^T\|_{2,1}$

Fig. 1. The visualization of the learned $\mathbf{W}$ on the FG-NET dataset. (a) Each row is the l_2 norm value of each row of $\mathbf{W}$. (b) Each column is the l_2 norm value of each column of $\mathbf{W}$. Dark blue denotes that the values are close to zero.

In order to solve (6), we first introduce three variables $\widehat{\mathbf{W}}$, $\widetilde{\mathbf{W}}$ and $\mathbf{Z}$, to convert (6) to the following equivalent objective function:

$$\begin{aligned}\min_{\mathbf{W},\widehat{\mathbf{W}},\widetilde{\mathbf{W}},\mathbf{Z}} & \|\mathbf{X}-\mathbf{X}\mathbf{W}\mathbf{X}\|_F^2+\alpha\|\widehat{\mathbf{W}}\|_{2,1}+\beta\|\widetilde{\mathbf{W}}\|_{2,1} \\ & +\lambda\|\mathbf{T}\odot\mathbf{Z}\|_1 \\ s.t.\ & \mathbf{W}\mathbf{X}=\mathbf{Z},\mathbf{W}=\widehat{\mathbf{W}},\mathbf{W}^T=\widetilde{\mathbf{W}}. \end{aligned} \tag{8}$$

The augmented Lagrange function of (8) is

$$\begin{aligned}&\mathcal{L}_{\rho_1,\rho_2,\rho_3}(\mathbf{W},\widehat{\mathbf{W}},\widetilde{\mathbf{W}},\mathbf{Z},\Lambda_1,\Lambda_2,\Lambda_3):=\|\mathbf{X}-\mathbf{X}\mathbf{W}\mathbf{X}\|_F^2 \\ &+\alpha\|\widehat{\mathbf{W}}\|_{2,1}+\beta\|\widetilde{\mathbf{W}}\|_{2,1}+\lambda\|\mathbf{T}\odot\mathbf{Z}\|_1+\langle\Lambda_1,\mathbf{W}\mathbf{X}-\mathbf{Z}\rangle \\ &+\langle\Lambda_2,\mathbf{W}-\widehat{\mathbf{W}}\rangle+\langle\Lambda_3,\mathbf{W}^T-\widetilde{\mathbf{W}}\rangle+\frac{\rho_1}{2}\|\mathbf{W}\mathbf{X}-\mathbf{Z}\|_F^2 \\ &+\frac{\rho_2}{2}\|\mathbf{W}-\widehat{\mathbf{W}}\|_F^2+\frac{\rho_3}{2}\|\mathbf{W}^T-\widetilde{\mathbf{W}}\|_F^2, \end{aligned} \tag{9}$$

where Λ_1, Λ_2 and Λ_3 are Lagrange multipliers. ρ_1, ρ_2 and ρ_3 are the constraint violation penalty parameters. From the augmented Lagrangian function, we can find that the subproblems about $\widehat{\mathbf{W}}$, $\widetilde{\mathbf{W}}$ and $\mathbf{Z}$ are totally separable, as a result we can introduce the classical two-block ADMM here, while considering $\mathbf{W}$ and $(\widehat{\mathbf{W}},\widetilde{\mathbf{W}},\mathbf{Z})$ as two-block variables. Recall that in a ADMM-type algorithm, the basic Gauss-Seidel structure in $(t+1)$-th iteration is as follows,

$$\begin{cases}\mathbf{W}^{k+1}=\arg\min\mathcal{L}(\mathbf{W},\widehat{\mathbf{W}}^k,\widetilde{\mathbf{W}}^k,\mathbf{Z}^k,\Lambda_1^k,\Lambda_2^k,\Lambda_3^k),\\ \widehat{\mathbf{W}}^{k+1}=\arg\min\mathcal{L}(\mathbf{W}^{k+1},\widehat{\mathbf{W}},\widetilde{\mathbf{W}}^k,\mathbf{Z}^k,\Lambda_1^k,\Lambda_2^k,\Lambda_3^k),\\ \widetilde{\mathbf{W}}^{k+1}=\arg\min\mathcal{L}(\mathbf{W}^{k+1},\widehat{\mathbf{W}}^{k+1},\widetilde{\mathbf{W}},\mathbf{Z}^k,\Lambda_1^k,\Lambda_2^k,\Lambda_3^k),\\ \mathbf{Z}^{k+1}=\arg\min\mathcal{L}(\mathbf{W}^{k+1},\widehat{\mathbf{W}}^{k+1},\widetilde{\mathbf{W}}^{k+1},\mathbf{Z},\Lambda_1^k,\Lambda_2^k,\Lambda_3^k),\\ \Lambda_1^{k+1}=\Lambda_1^k+\rho_1(\mathbf{W}^{k+1}\mathbf{X}-\mathbf{Z}^{k+1}),\\ \Lambda_2^{k+1}=\Lambda_2^k+\rho_2(\mathbf{W}^{k+1}-\widehat{\mathbf{W}}^{k+1}),\\ \Lambda_3^{k+1}=\Lambda_3^k+\rho_3((\mathbf{W}^{k+1})^T-\widetilde{\mathbf{W}}^{k+1}).\end{cases}$$

Next, we will introduce how to solve these subproblems in detail.

i) Compute the subproblem about $\mathbf{W}^{k+1}$: When the other variables are fixed with the former iteration result $(\widehat{\mathbf{W}}^k,\widetilde{\mathbf{W}}^k,\mathbf{Z}^k,\Lambda_1^k,\Lambda_2^k,\Lambda_3^k)$, the subproblem about $\mathbf{W}^{k+1}$ is as follows:

$$\begin{aligned}\mathbf{W}^{k+1}&=\arg\min_{\mathbf{W}}\mathcal{L}_{\rho_1,\rho_2,\rho_3}(\mathbf{W},\widehat{\mathbf{W}}^k,\widetilde{\mathbf{W}}^k,\mathbf{Z}^k,\Lambda_1^k,\Lambda_2^k,\Lambda_3^k)\\ &=\arg\min_{\mathbf{W}}\|\mathbf{X}-\mathbf{X}\mathbf{W}\mathbf{X}\|_F^2+\frac{\rho_1}{2}\|\mathbf{W}\mathbf{X}-\mathbf{Z}^k+\frac{\Lambda_1^k}{\rho_1}\|_F^2\\ &\quad+\frac{\rho_2}{2}\|\mathbf{W}-\widehat{\mathbf{W}}^k+\frac{\Lambda_2^k}{\rho_2}\|_F^2+\frac{\rho_3}{2}\|\mathbf{W}^T-\widetilde{\mathbf{W}}^k+\frac{\Lambda_3^k}{\rho_3}\|_F^2.\end{aligned}$$

The necessary optimality condition further follows as

$$\frac{\partial\mathcal{L}_{\rho_1,\rho_2,\rho_3}(\mathbf{W},\widehat{\mathbf{W}}^k,\widetilde{\mathbf{W}}^k,\mathbf{Z}^k,\Lambda_1^k,\Lambda_2^k,\Lambda_3^k)}{\partial\mathbf{W}}=0. \tag{10}$$

This implies

$$\begin{aligned}&(2\mathbf{X}^T\mathbf{X}+\rho_1\mathbf{I})\mathbf{W}\mathbf{X}\mathbf{X}^T+(\rho_2+\rho_3)\mathbf{W}=2\mathbf{X}^T\mathbf{X}\mathbf{X}^T\\ &+\rho_1(\mathbf{Z}^k-\frac{\Lambda_1^k}{\rho_1})\mathbf{X}^T+\rho_2(\widehat{\mathbf{W}}^k-\frac{\Lambda_2^k}{\rho_2})+\rho_3(\widetilde{\mathbf{W}}^k-\frac{\Lambda_3^k}{\rho_3})^T.\end{aligned}$$

For writing conveniently, let $\mathbf{M}=2\mathbf{X}^T\mathbf{X}+\rho_1\mathbf{I}$, and $\mathbf{H}=2\mathbf{X}^T\mathbf{X}\mathbf{X}^T+\rho_1(\mathbf{Z}^k-\frac{\Lambda_1^k}{\rho_1})\mathbf{X}^T+\rho_2(\widehat{\mathbf{W}}^k-\frac{\Lambda_2^k}{\rho_2})+\rho_3(\widetilde{\mathbf{W}}^k-\frac{\Lambda_3^k}{\rho_3})^T$, then the equation above becomes

$$\mathbf{M}\mathbf{W}\mathbf{X}\mathbf{X}^T+(\rho_2+\rho_3)\mathbf{W}=\mathbf{H}. \tag{11}$$

Since $\mathbf{M}$ and $\mathbf{X}\mathbf{X}^T$ are positive semi-definite and symmetric, we can do eigenvalue decomposition with all non-negative eigenvalues, obtaining

$$\begin{cases}\mathbf{M}=\mathbf{P}\Theta_1\mathbf{P}^T,\\ \mathbf{X}\mathbf{X}^T=\mathbf{Q}\Theta_2\mathbf{Q}^T,\end{cases} \tag{12}$$

where $\mathbf{P}$ and $\mathbf{Q}$ are both orthogonal. Θ_1 and Θ_2 are two diagonal matrices.

Plugging (12) into (11), we obtain

$$\begin{aligned}&\mathbf{P}\Theta_1\mathbf{P}^T\mathbf{W}\mathbf{Q}\Theta_2\mathbf{Q}^T+(\rho_2+\rho_3)\mathbf{W}=\mathbf{H}\\ \Rightarrow&\Theta_1\mathbf{P}^T\mathbf{W}\mathbf{Q}\Theta_2+(\rho_2+\rho_3)\mathbf{P}^T\mathbf{W}\mathbf{Q}=\mathbf{P}^T\mathbf{H}\mathbf{Q}.\end{aligned} \tag{13}$$

Let $\mathbf{Y}=\mathbf{P}^T\mathbf{W}\mathbf{Q}$, then (13) becomes

$$\begin{aligned}&\Theta_1\mathbf{Y}\Theta_2+(\rho_2+\rho_3)\mathbf{Y}=\mathbf{P}^T\mathbf{H}\mathbf{Q}\\ \Rightarrow&\mathbf{Y}_{ij}=\frac{(\mathbf{P}^T\mathbf{H}\mathbf{Q})_{ij}}{(\Theta_1)_{ii}(\Theta_2)_{jj}+\rho_2+\rho_3},i=1,\ldots,n,j=1,\ldots,d.\end{aligned}$$

As we know, $(\Theta_1)_{ii}\geq 0,(\Theta_2)_{jj}\geq 0$. In the meantime, ρ_2 and ρ_3 are greater than zero in practice, so the denominator in the equation above is greater than zero. After obtaining $\mathbf{Y}$, we can easily calculate $\mathbf{W}^{k+1}$ as

$$\mathbf{W}^{k+1}=\mathbf{P}\mathbf{Y}\mathbf{Q}^T \tag{14}$$

ii) Further we calculate the subproblem about $\widehat{\mathbf{W}}^{k+1}$, i.e.,

$$\begin{aligned}\widehat{\mathbf{W}}^{k+1}&=\arg\min_{\widehat{\mathbf{W}}}\mathcal{L}(\mathbf{W}^{k+1},\widehat{\mathbf{W}},\widetilde{\mathbf{W}}^k,\mathbf{Z}^k,\Lambda_1^k,\Lambda_2^k,\Lambda_3^k)\\ &=\arg\min_{\widehat{\mathbf{W}}}\alpha\|\widehat{\mathbf{W}}\|_{2,1}+\frac{\rho_2}{2}\|\widehat{\mathbf{W}}-\mathbf{W}^{k+1}-\frac{\Lambda_2^k}{\rho_2}\|_F^2.\end{aligned} \tag{15}$$

In order to solve the subproblem (15), we first decouple it as:

$$\begin{aligned}\widehat{\mathbf{W}}^{k+1}=&\arg\min_{\widehat{\mathbf{W}}^i}\sum_{i=1}^n\alpha\|\widehat{\mathbf{W}}^i\|_2\\ &+\frac{\rho_2}{2}\sum_{i=1}^n\|\widehat{\mathbf{W}}^i-(\mathbf{W}^{k+1}+\frac{\Lambda_2^k}{\rho_2})^i\|_2^2,\end{aligned} \tag{16}$$

where $\widehat{\mathbf{W}}^i$ and $(\mathbf{W}^{k+1}+\frac{1}{\rho_2}\Lambda_2^k)^i$ are the i-th row of matrix $\widehat{\mathbf{W}}$ and $\mathbf{W}^{k+1}+\frac{1}{\rho_2}\Lambda_2^k$ respectively. The problem (16) can be solved by the following lemma [35]:

Lemma 2.1. For any $\sigma,\eta>0$, and $\mathbf{v}\in\mathbb{R}^q$, the minimizer of

$$\min_{\mathbf{u}\in\mathbb{R}^q}\sigma\|\mathbf{u}\|_2+\frac{\eta}{2}\|\mathbf{u}-\mathbf{v}\|_2^2, \tag{17}$$

is given by

$$\mathbf{u}=\begin{cases}(1-\frac{\sigma}{\eta\|\mathbf{v}\|_2})\mathbf{v}, & \|\mathbf{v}\|_2>\frac{\sigma}{\eta}\\ 0, & \|\mathbf{v}\|_2\le\frac{\sigma}{\eta}.\end{cases} \tag{18}$$

Based on this lemma, we can obtain the optimal $\widehat{\mathbf{W}}^{k+1}$ as

$$(\widehat{\mathbf{W}}^{k+1})^i=\begin{cases}(1-\frac{\alpha}{\rho_2\|\mathbf{s}\|_2})\mathbf{s}, & \|\mathbf{s}\|_2>\frac{\alpha}{\rho_2}\\ 0, & \|\mathbf{s}\|_2\le\frac{\alpha}{\rho_2},\end{cases} \tag{19}$$

where $\mathbf{s}=(\mathbf{W}^{k+1}+\frac{1}{\rho_2}\Lambda_2^k)^i$.

iii) $\widetilde{\mathbf{W}}^{k+1}$ is the minimizer for

$$\begin{aligned}&\min_{\widetilde{\mathbf{W}}}\mathcal{L}(\mathbf{W}^{k+1},\widehat{\mathbf{W}}^{k+1},\widetilde{\mathbf{W}},\mathbf{Z}^k,\Lambda_1^k,\Lambda_2^k,\Lambda_3^k)\\ =&\min_{\widetilde{\mathbf{W}}}\beta\|\widetilde{\mathbf{W}}\|_{2,1}+\frac{\rho_3}{2}\|\widetilde{\mathbf{W}}-\left((\mathbf{W}^{k+1})^T+\frac{\Lambda_3^k}{\rho_3}\right)\|_F^2\end{aligned} \tag{20}$$

Similar to solve (15), the optimal $\widetilde{\mathbf{W}}^{k+1}$ can be easily obtained by

$$(\widetilde{\mathbf{W}}^{k+1})^i=\begin{cases}(1-\frac{\beta}{\rho_3\|\mathbf{s}\|_2})\mathbf{s}, & \|\mathbf{s}\|_2>\frac{\beta}{\rho_3}\\ 0, & \|\mathbf{s}\|_2\le\frac{\beta}{\rho_3},\end{cases} \tag{21}$$

where $\mathbf{s}=\left((\mathbf{W}^{k+1})^T+\frac{1}{\rho_3}\Lambda_3^k\right)^i$.

iv) In order to compute the subproblem about $\mathbf{Z}^{k+1}$, we need to solve

$$\begin{aligned}&\min_{\mathbf{Z}}(\mathbf{W}^{k+1},\widehat{\mathbf{W}}^{k+1},\widetilde{\mathbf{W}}^{k+1},\mathbf{Z},\Lambda_1^k,\Lambda_2^k,\Lambda_3^k)\\ =&\min_{\mathbf{Z}}\lambda\|\mathbf{T}\odot\mathbf{Z}\|_1+\frac{\rho_1}{2}\|\mathbf{Z}-\mathbf{W}^{k+1}\mathbf{X}-\frac{\Lambda_1^k}{\rho_1}\|_F^2.\end{aligned} \tag{22}$$

The problem (22) can be solved by the following matrix shrinkage operation Lemma [36]:

Lemma 2.2. For $\mu>0$, and $\mathbf{K}\in\mathbb{R}^{s\times t}$, the solution of the problem

$$\min_{\mathbf{L}\in\mathbb{R}^{s\times t}}\mu\|\mathbf{L}\|_1+\frac{1}{2}\|\mathbf{L}-\mathbf{K}\|_F^2,$$

is given by $L_\mu(\mathbf{K})\in\mathbb{R}^{s\times t}$, which is defined componentwisely by

$$(L_\mu(\mathbf{K}))_{ij}:=\max\{|\mathbf{K}_{ij}|-\mu,0\}\cdot sgn(\mathbf{K}_{ij}), \tag{23}$$

where $sgn(t)$ is the signum function of $t\in\mathbf{R}$, i.e.,

$$sgn(t):=\begin{cases}+1 & \text{if } t>0,\\ 0 & \text{if } t=0,\\ -1 & \text{if } t<0.\end{cases}$$

Based on Lemma 2.2, we can obtain a closed-form solution of $\mathbf{Z}^{k+1}$ whose (i,j)-th entry is expressed as

$$\begin{aligned}\mathbf{Z}_{ij}^{k+1}:=&\max\{|(\mathbf{W}^{k+1}\mathbf{X}+\frac{\Lambda_1^k}{\rho})_{ij}|-\frac{\lambda\cdot\mathbf{T}_{ij}}{\rho_1},0\}\\ &\cdot sgn((\mathbf{W}^{k+1}\mathbf{X}+\frac{\Lambda_1^k}{\rho_1})_{ij}).\end{aligned} \tag{24}$$

The key steps of the proposed ALFS algorithm are summarized in Algorithm 1. We can also extend our method to the kernel version by defining a new data representation to incorporate the kernel information as in [37].

Algorithm 1 The ALFS Algorithm

Input: The data matrix $\mathbf{X}\in\mathbb{R}^{d\times n}$, parameters α, β, and λ.

Initialize: $\mathbf{W}^0=\widehat{\mathbf{W}}^0=\mathbf{0}$, $\widetilde{\mathbf{W}}^{\mathbf{0}}=0$, $\mathbf{Z}^0=\mathbf{0}$, $\Lambda_1^0=\mathbf{0}$, $\Lambda_2^0=\mathbf{0}$, $\Lambda_3^0=\mathbf{0}$, $\rho_1=\rho_2=\rho_3=10^{-6}$, $\max_\rho=10^{10}$, $\tau=1.1$, $\epsilon=10^{-3}$, $k=0$.

while not converged **do**

1. fix the other variables and update $\mathbf{W}^{k+1}$ by (14);
2. fix the other variables and update $\widehat{\mathbf{W}}^{k+1}$ by (19);
3. fix the other variables and update $\widetilde{\mathbf{W}}^{k+1}$ by (21);
4. fix the other variables and update $\mathbf{Z}^{k+1}$ by (24);
5. update the multipliers
   $\Lambda_1^{k+1}=\Lambda_1^k+\rho_1(\mathbf{W}^{k+1}\mathbf{X}-\mathbf{Z}^{k+1})$,
   $\Lambda_2^{k+1}=\Lambda_2^k+\rho_2(\mathbf{W}^{k+1}-\widehat{\mathbf{W}}^{k+1})$,
   $\Lambda_3^{k+1}=\Lambda_3^k+\rho_3\left((\mathbf{W}^{k+1})^T-\widetilde{\mathbf{W}}^{k+1}\right)$;
6. update the parameters ρ_1, ρ_2, and ρ_3 by
   $\rho_1=\min(\tau\rho_1,\max_\rho)$,
   $\rho_2=\min(\tau\rho_2,\max_\rho)$,
   $\rho_3=\min(\tau\rho_3,\max_\rho)$;
7. $k\leftarrow k+1$;
8. check the convergence conditions
   $\|\mathbf{W}^k\mathbf{X}-\mathbf{Z}^k\|_\infty<\epsilon$ and $\|\mathbf{W}^k-\widehat{\mathbf{W}}^k\|_\infty<\epsilon$ and
   $\|(\mathbf{W}^k)^T-\widetilde{\mathbf{W}}^k\|_\infty<\epsilon$ and $|\frac{f(\mathbf{W}^k)-f(\mathbf{W}^{k-1})}{f(\mathbf{W}^{k-1})}|<\epsilon$, where $f(\mathbf{W}^k)$ is the objective function value of (6) at the point $\mathbf{W}^k$.

end while

Output: The matrix $\mathbf{W}^k\in\mathbb{R}^{n\times d}$.

2.5 Algorithm Analysis

From the framework of ALFS, we can find that Algorithm 1 is the direct application of the classical two-block ADMM, although the problem has more than two block variables. All the subproblems in Algorithm 1 have closed-form solutions. Based on the classical convergence results, we can obtain the global convergence of Algorithm 1 to the primal-dual optimal solution of problem (8) (see [38], [39]). In the following we present both the global convergence and the iteration complexity results of Algorithm 1.

Theorem 2.1. For given constant parameters α, β, γ and given constant penalty parameters ρ_1, ρ_2, ρ_3. Denote the iteration sequence generated by Algorithm 1 as

$$\begin{cases}\mathbf{\Sigma}^k:=\left\{\mathbf{W}^k,\widehat{\mathbf{W}}^k,\widetilde{\mathbf{W}}^k,\mathbf{Z}^k,\Lambda_1^k,\Lambda_2^k,\Lambda_3^k\right\},\\ \tilde{\mathbf{\Sigma}}^k:=\frac{1}{k+1}\sum_{t=0}^k\mathbf{\Sigma}^t,\\ \mathbf{\Sigma}_1^k:=\left\{\mathbf{W}^k,\widehat{\mathbf{W}}^k,\widetilde{\mathbf{W}}^k,\mathbf{Z}^k\right\},\\ \mathbf{\Sigma}_2^k:=\left\{\Lambda_1^k,\Lambda_2^k,\Lambda_3^k\right\}.\end{cases}$$

Then we have the following results:

1) (Global Convergence) The sequence $\{\mathbf{\Sigma}^k\}$ converges to a primal-dual optimal solution pair $(\mathbf{W}^\infty,\widehat{\mathbf{W}}^\infty,\widetilde{\mathbf{W}}^\infty,\mathbf{Z}^\infty,\Lambda_1^\infty,\Lambda_2^\infty,\Lambda_3^\infty)$, where $(\mathbf{W}^\infty,\widehat{\mathbf{W}}^\infty,\widetilde{\mathbf{W}}^\infty,\mathbf{Z}^\infty)$ is the global optimal solution of problem (8) and $\mathbf{W}^\infty$ is the global optimal solution of problem (6).
2) (Constraint Satisfactory) Both constraint violations will converge to zero, e.g.

$$\begin{cases}\left\|\mathbf{W}^k\mathbf{X}-\mathbf{Z}^k\right\|_F\to 0,\\ \left\|\mathbf{W}^k-\widehat{\mathbf{W}}^k\right\|_F\to 0,\\ \left\|(\mathbf{W}^k)^T-\widetilde{\mathbf{W}}^k\right\|_F\to 0.\end{cases}$$

3) (Ergodic Iteration Complexity [40]) Let $(\mathbf{W}^*,\widehat{\mathbf{W}}^*,\widetilde{\mathbf{W}}^*,\mathbf{Z}^*,\Lambda_1^*,\Lambda_2^*,\Lambda_3^*)$ be an optimal solution pair, we have

$$\mathcal{L}_{\rho_1,\rho_2,\rho_3}(\tilde{\mathbf{\Sigma}}_1^k,\mathbf{\Sigma}_2^*)-\mathcal{L}_{\rho_1,\rho_2,\rho_3}(\mathbf{\Sigma}_1^*,\tilde{\mathbf{\Sigma}}_2^k)\leq\frac{C_1}{k+1}, \tag{25}$$

where C_1 denotes a constant related with $\mathbf{\Sigma}^0$ and $\mathbf{\Sigma}^*$.

4) (Non-ergodic Iteration Complexity [41]) The non-ergodic iteration complexity can be written as

$$\left\|\mathbf{\Sigma}^k-\mathbf{\Sigma}^{k+1}\right\|_{\mathcal{H}}^2\leq\frac{C_2}{k+1}, \tag{26}$$

where C_2 also denotes a constant related with $\mathbf{\Sigma}^0$ and $\mathbf{\Sigma}^*$ and $\mathcal{H}$ is a matrix related with $\mathbf{X}$ as follows,

$$\mathcal{H}=\begin{pmatrix}\mathbf{S}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}\\ \mathbf{0}&\rho_2\mathbf{I}&\ddots&\ddots&\ddots&\ddots&\mathbf{0}\\ \mathbf{0}&\ddots&\rho_3\mathbf{I}&\ddots&\ddots&\ddots&\mathbf{0}\\ \mathbf{0}&\ddots&\ddots&\rho_1\mathbf{I}&\ddots&\ddots&\mathbf{0}\\ \mathbf{0}&\ddots&\ddots&\ddots&\frac{1}{\rho_1}\mathbf{I}&\ddots&\mathbf{0}\\ \mathbf{0}&\ddots&\ddots&\ddots&\ddots&\frac{1}{\rho_2}\mathbf{I}&\mathbf{0}\\ \mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\frac{1}{\rho_3}\mathbf{I}\end{pmatrix},$$

where $\mathbf{S}=\rho_1\mathbf{X}^T\mathbf{X}+(\rho_2+\rho_3)\mathbf{I}$.

For the detailed proof, please refer to [39], [40]. We do not present them here to save space. The first and second parts of this theorem show the global convergence of the presented algorithm, including sequence convergence and constraint convergence. From the first part, we can find that the sequence converges to the primal-dual optimal solution pair, while the second part shows the two linear constraints converge to zero in the sense of *Frobenius* norm.

The third and fourth parts above show a global convergence speed of ADMM, in the sense of ergodic and non-ergodic respectively. The inequality (25) is the ergodic iteration complexity, which denotes the characterization of ϵ-optimal based on primal-dual optimality gap as follows,

$$\begin{aligned}\text{Gap}(\mathbf{\Sigma}_1,\mathbf{\Sigma}_2)&:=\mathcal{L}_{\rho_1,\rho_2}(\mathbf{\Sigma}_1,\mathbf{\Sigma}_2^*)-\mathcal{L}_{\rho_1,\rho_2}(\mathbf{\Sigma}_1^*,\mathbf{\Sigma}_2)\\&\leq\epsilon.\end{aligned} \tag{27}$$

Thus, it means that after k iterations, we can obtain an $\mathcal{O}(1/k)$-optimal solution. The inequality (26) calculates the optimality condition between adjacent iterations, although this can not indicate convergence, but it really can accelerate the global convergence.

The above theorem not only shows the global convergence of Algorithm 1, but also presents two cases of iteration complexity. The global convergence means that the generated sequence converges to the optimal solution based on any initial point. Further the iteration complexity results mean that how good the iteration result is after k iterations. We can also find that both iteration complexity results are $O(1/k)$, which is in the same order with many first-order algorithms.

TABLE 1
Summary of experimental datasets. 'SP', 'FT', 'CT' denote the number of samples, the number of features, and the number of categories, respectively.

Dataset	SP	FT	CT	Application	Type
Madelon	2600	500	2	Feature Selection Challenge	Artificial Data
TOX-171	171	5748	4	Toxicity Prediction	Microarray
Musk	476	168	2	Musk Activity Prediction	Microarray
ORL	400	512	40	Face Recognition	Image
FG-NET	1002	907	5	Age Estimation	Image
UCF11	1600	512	11	Action Recognition	Video

3 RELATED WORK

As described, the work most related to our proposed approach is the second group active learning methods that intend to select the most representative samples. In this section, we will briefly provide a review of the approaches in this group. Among them, the most popular one is the Transductive Experimental Design (TED) [18]. TED aimed to find a representative sample subset from the unlabeled dataset, such that the dataset can be best approximated by linear combinations of the selected samples. Since this optimization problem is NP-hard, [18] proposed a suboptimal sequential optimization algorithm and a non-greedy optimization algorithm to solve it, respectively.

Following TED, more active learning algorithms have been developed. Cai and He [20] extended TED to choose samples by utilizing a nearest neighbor graph to capture intrinsic local manifold structure, where the graph Laplacian is incorporated into a manifold adaptive kernel space. Zhang et al. [42] adopted the idea from Locally Linear Embedding (LLE) [43] to find the reconstruction coefficients. They represented each sample by a linear combination of its neighbors, which can well preserve the local geometrical structure of the data. Similar to [42], Hu et al. [21] incorporated the local geometrical information into the active learning process. Specifically, they introduced a regularization term to make the nearer neighbors have much effect on the linear reconstruction of data point, and penalized the selected samples distant from the reconstructed sample severely. Nie et al. [17] proposed a novel method to relax the objective of TED to an efficient convex formulation, and utilized the robust sparse representation loss function to reduce the effect of outliers.

4 EXPERIMENT

In this section, we empirically evaluate the proposed method, ALFS, on six publicly available datasets, including artificial data, microarray data, image data, and video data.

4.1 Experimental Setting

Dataset We evaluate the performance of our ALFS on six datasets, including 1) an artificial dataset, Madelon [44],

used in the NIPS 2003 feature selection challenge; 2) two preprocessed microarray datasets, TOX-171 [45] and Musk [44], for studying toxicity prediction and musk activity prediction respectively; 3) two image datasets, ORL and FG-NET, which are widely used for face recognition [46] and facial age estimation [47], [48], respectively. For FG-NET, we conduct age range estimation as in [49], and divide the dataset into five age groups in the experiment; 4) a video dataset, UCF11 [50], which is collected from YouTube. It contains 11 video action categories, which are very challenging to be recognized due to large variations in camera motion, object appearance, pose, object scale, viewpoint, cluttered background, and illumination conditions. Datasets from different areas serve as a good test bed for a comprehensive evaluation. Table 1 summarizes the details of the datasets used in the experiments.

Compared methods Since ALFS is related to the second group of active learning algorithms, we compare it with some state-of-the-art approaches in this group to demonstrate the effectiveness of ALFS. The compared methods in the experiments are listed as follows:

- Random Sampling (RS): randomly selects samples from the candidate dataset, which is used as the baseline for active learning.
- Transductive Experimental Design (TED) [18][3]: an active learning method developing experimental design in a transductive setting.
- RRSS [17][4]: an active learning method taking advantage of robust representation and structure sparsity.
- Active Learning via Neighbor Reconstruction (ALNR) [21]: an active learning method using neighborhood reconstruction.
- R-CUR [30]: a randomized algorithmic approach for solving CUR matrix factorization. We name it R-CUR for short.
- ALFS-I: our proposed method for selecting representative sample and feature selection simultaneously, but without considering local linear reconstruction.
- ALFS-II: our proposed method for simultaneous active learning and feature selection with local linear reconstruction.

To further show the benefit of simultaneous active sample selection and feature selection, we also compare our ALFS against some feature selection approaches combined with the active learning approaches above, i.e., first using feature selection methods to reduce the dimensionality, and then applying the active learning methods above to select samples based on the new low-dimensional representation. We use three kinds of unsupervised feature selection methods, Laplacian[5] [51], SPEC[6] [22], and SOGFS[7] [46], to combine with the active learning algorithms in the experiments, respectively.

Experiments protocol Following [17], for each dataset, we first randomly select 50% of the data points as candidate samples for training, from which we apply the compared active learning methods to select a subset of samples to request human labeling. Using the selected samples and their queried labels as training data, we learn a classification model, and evaluate the representativeness of the selected samples in terms of classification accuracy on the rest 50% data samples. The latter is regarded as the testing data. In order to demonstrate that our method is not sensitive to different classifiers, we use two kinds of classical classification models, support vector machine (SVM) and decision tree, to evaluate the effectiveness of the proposed method, respectively. For simplicity, we use the linear kernel in SVM, and fix the hyperparameter $C = 100$ through the experiments. The parameters α, β, and λ in our algorithm are searched from $\{10^{-4}, 10^{-3}, \ldots, 10^{0}, \ldots, 10^{3}\}$. For a fair comparison, the parameters in TED, RRSS, and ALNR are also searched from the same space. In the experiment, we repeat every test case for 10 times, and report the average classification performance.

4.2 Experimental Result

Comparison with Active Learning Algorithms In order to demonstrate the effectiveness of our ALFS in selecting representative samples, we compare ALFS with some state-of-the-art active learning algorithms. For ALFS and R-CUR, we vary the number of selected features from 10 to 100 with an incremental step of 10 on all the datasets[8], and report the best results. The results are shown in Fig. 2 and Fig. 3. We can observe that our ALFS-I obtains better performance than all the other candidates on all the datasets, which shows that joint active learning with feature selection is beneficial to improving classification accuracy. ALFS-II achieves the best classification performance among all the datasets under different classifiers. On certain datasets, such as the Madelon dataset, our ALFS-I and ALFS-II are significantly better than the other methods. For the Madelon dataset, when the number of the selected samples is set to 1200 and using SVM as the classifier, ALFS-I and ALFS-II obtain 14.8% and 15.4% relative improvement over the second best result, i.e., ALNR, respectively. When combined with decision tree, ALFS-I and ALFS-II attain 11.3% and 12.3% relative improvement over ALNR, respectively. In addition, ALFS-II outperforms ALFS-I, which means that incorporating local linear reconstruction into the process of active learning and feature selection is helpful for improving performance. Moreover, we also observe some other interesting phenomenon. First, we note that R-CUR does not show its competency, compared to our ALFS-I and ALFS-II on almost all the datasets. The reason is that R-CUR [30] is a general CUR model and adopts a randomized algorithmic approach to seek the matrices $\mathbf{C}$ and $\mathbf{R}$ for satisfying (1). It does not consider it as an optimization problem, making the selected samples and features hardly be the most representative,

3. A sequential solver can be downloaded from http://www.dbs.ifi.lmu.de/~yu_k/ted/.

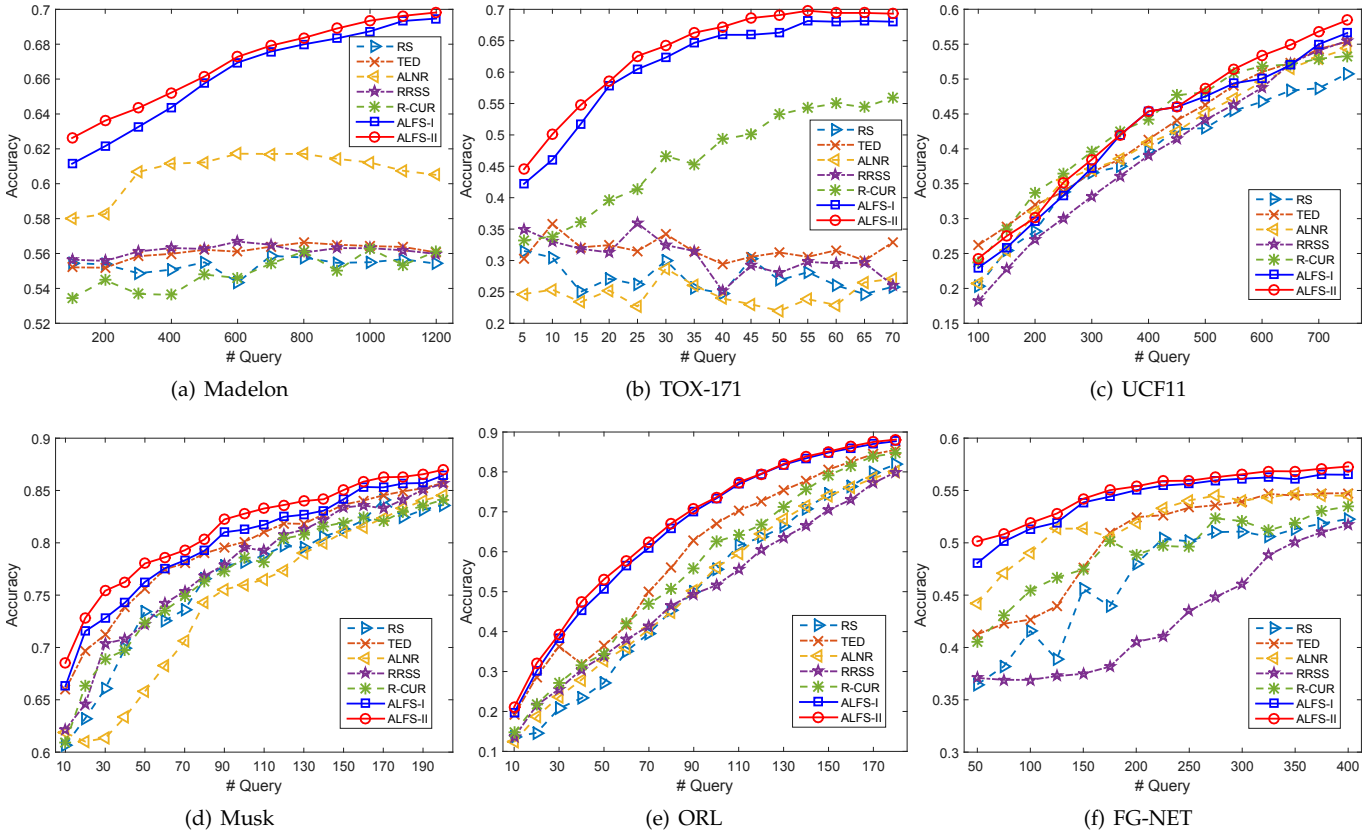
4. The code can be downloaded from http://www.escience.cn/people/fpnie/papers.html.

5. We downloaded the code from http://www.cad.zju.edu.cn/home/dengcai/Data/MCFS.html.

6. The code can be downloaded from http://featureselection.asu.edu/software.php.

7. The code can be downloaded from http://www.escience.cn/people/fpnie/index.html.

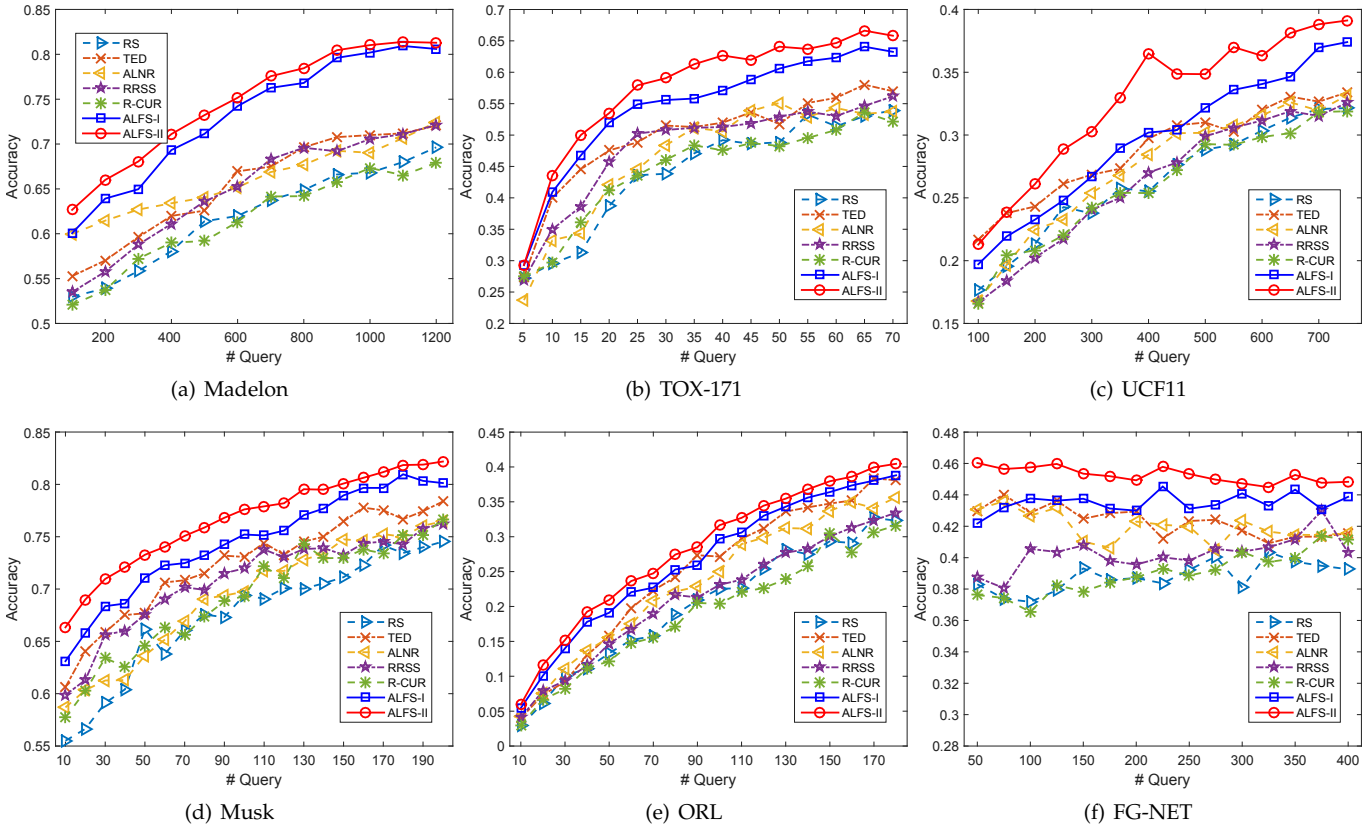
8. When the user inputs the desired number of samples m and the number of features r, the final outputs m' and r' of R-CUR [30] may be slightly different m and r, respectively.



(a) Madelon (b) TOX-171 (c) UCF11

(d) Musk (e) ORL (f) FG-NET

Fig. 2. Comparisons of different active learning methods combined with the SVM classifier on six benchmark datasets. The curve shows the learning accuracy over queries.



(a) Madelon (b) TOX-171 (c) UCF11

(d) Musk (e) ORL (f) FG-NET

Fig. 3. Comparisons of different active learning methods combined with the decision tree classifier on six benchmark datasets. The curve shows the learning accuracy over queries.

TABLE 2
Accuracy (%) of feature selection + active learning algorithms on the Madelon dataset. Best results in each column are highlighted in bold fonts.

(a) SVM

Method	#Dim					
	10	30	50	70	90	500
Laplacian+TED	57.2	59.4	58.8	58.2	56.6	56.1
SPEC+TED	54.3	54.8	56.0	56.0	54.7	56.1
SOGFS+TED	49.9	51.9	53.1	55.1	57.2	56.1
Laplacian+RRSS	64.3	63.7	61.2	60.0	58.7	56.0
SPEC+RRSS	62.8	59.8	58.6	58.0	57.5	56.0
SOGFS+RRSS	48.6	56.2	59.2	60.0	60.7	56.0
Laplacian+ALNR	62.1	61.6	61.1	60.6	60.3	**60.5**
SPEC+ALNR	60.2	59.3	58.8	59.0	58.9	60.5
SOGFS+ALNR	49.5	50.7	50.4	51.1	59.3	60.5
R-CUR	49.5	54.4	52.0	56.1	54.1	55.4
ALFS-I	69.5	64.2	61.9	61.5	61.0	56.7
ALFS-II	**69.8**	**64.8**	**62.7**	**62.2**	**61.8**	57.2

(b) Decision Tree

Method	#Dim					
	10	30	50	70	90	500
Laplacian+TED	56.4	59.8	63.7	67.7	70.6	72.1
SPEC+TED	52.0	54.8	57.2	61.0	64.0	72.1
SOGFS+TED	51.5	54.1	55.2	58.0	61.4	72.1
Laplacian+RRSS	70.0	78.2	76.5	76.1	74.3	72.1
SPEC+RRSS	66.7	63.1	63.3	65.9	65.0	72.1
SOGFS+RRSS	50.9	54.7	60.4	65.3	68.3	72.1
Laplacian+ALNR	70.2	79.3	77.0	76.0	75.4	72.4
SPEC+ALNR	67.5	64.9	65.0	66.5	66.0	72.4
SOGFS+ALNR	51.4	51.9	52.1	53.1	62.8	72.4
R-CUR	50.3	52.4	52.0	55.6	53.7	67.9
ALFS-I	80.6	78.9	77.8	76.6	75.3	72.9
ALFS-II	**81.3**	**79.9**	**79.1**	**78.5**	**77.4**	**74.7**

TABLE 3
Accuracy (%) of feature selection + active learning algorithms on the TOX-171 dataset. Best results in each column are highlighted in bold fonts.

(a) SVM

Method	#Dim					
	10	30	50	70	90	5748
Laplacian+TED	48.5	58.1	60.8	60.7	60.7	32.9
SPEC+TED	27.8	30.2	29.4	30.0	29.3	32.9
SOGFS+TED	54.6	60.1	63.1	62.4	62.2	32.9
Laplacian+RRSS	52.0	59.0	60.6	59.2	59.1	26.1
SPEC+RRSS	52.0	59.0	60.6	59.2	59.1	26.1
SOGFS+RRSS	54.6	60.2	62.3	62.1	62.5	26.1
Laplacian+ALNR	51.5	55.6	58.4	57.3	56.2	27.1
SPEC+ALNR	25.8	26.5	24.0	24.9	24.1	27.1
SOGFS+ALNR	54.1	56.6	60.3	57.3	59.0	27.1
R-CUR	40.4	50.5	53.4	53.4	55.5	**41.3**
ALFS-I	53.7	66.5	67.6	67.4	67.6	36.6
ALFS-II	**58.4**	**68.0**	**69.3**	**69.0**	**68.5**	40.7

(b) Decision Tree

Method	#Dim					
	10	30	50	70	90	5748
Laplacian+TED	45.6	50.4	52.7	56.3	57.2	56.7
SPEC+TED	34.1	39.1	43.5	42.4	43.6	56.7
SOGFS+TED	47.5	50.3	51.9	55.4	56.4	56.7
Laplacian+RRSS	47.4	51.5	52.8	54.3	57.7	56.2
SPEC+RRSS	42.2	46.5	48.6	45.6	46.3	56.2
SOGFS+RRSS	45.1	50.5	50.5	53.4	53.3	56.2
Laplacian+ALNR	45.6	52.8	50.2	52.0	57.9	53.7
SPEC+ALNR	42.4	45.0	47.1	44.9	45.1	53.7
SOGFS+ALNR	43.7	50.3	53.7	52.4	55.1	53.7
R-CUR	43.1	45.9	50.8	50.6	51.3	52.1
ALFS-I	56.5	61.4	59.7	60.6	63.3	62.7
ALFS-II	**57.8**	**62.8**	**62.8**	**64.3**	**64.8**	**65.8**

which limits R-CUR to be directly applied to active learning and feature selection. Second, on the Madelon dataset and the TOX-171 dataset, the state-of-the-art active learning methods, ALNR, RRSS, TED, have poor classification performance, and do not improve the accuracy apparently as the number of the selected samples to be labeled increases, when use SVM as the final classifier. This is because that for the Madelon dataset, there are lots of noisy features (Based on the introduction in the webpage[9], there are only 5 informative features, 15 linear combinations of these five features, and 480 distractor features having no predictive power.), and for the TOX-171 dataset, there are 5748 features, being relatively large to the number of the samples. Thus, when using these features to train SVM without dimension reduction or removing noise variables, the model is very easy to overfit, which degrades the generality ability of the model. This also indicates that active learning can benefit from feature selection. In contrast, when combined with decision tree, these active learning methods obtain good results on these two datasets. The reason is that decision tree only employs a small subset of features to from the decision hyperplane, which can play the role of feature selection for removing noisy or redundant features to some extent. Third, active learning methods perform better performance than random sampling in general, especially when combined with decision tree. This shows that it is indeed meaningful to learn to select samples for human labeling in the scenario of supervised learning.

Comparison with Feature Selection + Active Learning In order to demonstrate the necessary of simultaneous sample and feature selection, we compare ALFS with some unsupervised feature selection methods combined with the active learning algorithms above. We fix the number of the selected samples to the truncations as shown in Fig. (2), and test the classification accuracies with different feature dimensions. The results are reported in Table 2-7. We can see that when using SVM or decision tree as the classifier, both of our ALFS-I and ALFS-II outperform those approaches treating sample selection and feature selection as two separate steps. Still taking the Madelon dataset as an example, when the number of selected features is set to 10, ALFS-II achieves 8.0% relative improvement over RRSS combined with Laplacian and SVM, 10.7% relative improvement over RRSS with SPEC and SVM, 15.1% relative improvement over RRSS with Laplacian and decision tree, and 20.8% relative improvement over RRSS with SPEC and decision tree, respectively. This further indicates that simultaneous sample and feature selection is promising for obtaining better performance. In addition, ALFS-II achieves better results than ALFS-I under various dimensions, which can come to the same conclusion as mentioned above. In the meantime, we also observe that our method usually has competitive results at the lower dimensions, and even has higher accuracies than using all the features under most of the datasets. It also verifies that it is meaningful to simultaneous active learning and feature selection.

9. https://archive.ics.uci.edu/ml/datasets/Madelon

TABLE 4
Accuracy (%) of feature selection + active learning algorithms on the UCF11 dataset. Best results in each column are highlighted in bold fonts.

(a) SVM

Method	#Dim					
	10	30	50	70	90	512
Laplacian+TED	29.1	40.9	45.2	50.4	55.5	55.3
SPEC+TED	21.5	33.2	41.0	48.6	54.3	55.3
SOGFS+TED	29.9	40.7	45.1	48.9	55.0	55.3
Laplacian+RRSS	36.7	48.3	51.6	53.3	55.5	55.5
SPEC+RRSS	29.5	40.8	46.1	50.3	54.5	55.5
SOGFS+RRSS	39.4	46.7	51.3	53.7	55.3	55.5
Laplacian+ALNR	35.8	47.1	50.3	52.5	54.5	54.4
SPEC+ALNR	29.3	41.1	46.5	50.2	53.5	54.4
SOGFS+ALNR	38.1	46.7	49.5	52.2	54.1	54.4
R-CUR	36.4	47.7	52.0	52.5	53.3	53.3
ALFS-I	44.8	51.8	53.7	56.3	56.4	56.6
ALFS-II	**45.3**	**52.8**	**55.3**	**57.1**	**58.4**	**58.2**

(b) Decision Tree

Method	#Dim					
	10	30	50	70	90	512
Laplacian+TED	25.6	30.0	31.5	32.5	33.2	33.4
SPEC+TED	18.4	25.6	28.3	30.5	31.5	33.4
SOGFS+TED	2.8	29.4	31.0	31.8	33.4	33.4
Laplacian+RRSS	35.3	35.6	31.6	32.6	32.4	32.7
SPEC+RRSS	24.9	26.8	29.0	30.1	31.2	32.7
SOGFS+RRSS	33.1	33.0	32.6	33.1	32.4	32.7
Laplacian+ALNR	34.0	35.0	33.9	32.7	32.6	33.2
SPEC+ALNR	25.1	26.7	28.7	29.3	31.8	33.2
SOGFS+ALNR	33.3	34.6	32.8	33.6	32.3	33.2
R-CUR	29.0	30.5	30.9	31.5	31.3	31.9
ALFS-I	37.4	36.2	36.1	35.3	35.2	34.9
ALFS-II	**39.1**	**39.1**	**38.2**	**38.0**	**36.8**	**36.9**

TABLE 5
Accuracy (%) of feature selection + active learning algorithms on the Musk dataset. Best results in each column are highlighted in bold fonts.

(a) SVM

Method	#Dim					
	10	30	50	70	90	168
Laplacian+TED	60.3	66.8	71.0	75.9	80.3	85.7
SPEC+TED	65.1	76.2	81.6	83.8	84.7	85.7
SOGFS+TED	68.1	70.6	73.7	78.2	82.3	85.7
Laplacian+RRSS	63.6	67.4	72.4	75.6	80.7	85.6
SPEC+RRSS	71.3	78.1	81.1	83.8	84.8	85.6
SOGFS+RRSS	66.9	69.4	75.0	79.5	83.2	85.6
Laplacian+ALNR	63.1	67.0	73.9	74.4	79.5	84.5
SPEC+ALNR	72.6	78.3	80.7	82.3	83.7	84.5
SOGFS+ALNR	64.5	71.6	72.8	75.3	79.9	84.5
R-CUR	71.9	76.7	80.8	82.0	83.7	84.1
ALFS-I	74.4	81.2	83.9	84.9	85.6	86.4
ALFS-II	**77.4**	**83.3**	**85.6**	**86.2**	**86.4**	**87.0**

(b) Decision Tree

Method	#Dim					
	10	30	50	70	90	168
Laplacian+TED	61.0	66.6	69.5	75.7	78.1	78.4
SPEC+TED	62.3	72.3	74.8	76.3	77.0	78.4
SOGFS+TED	71.6	72.7	75.1	77.3	78.6	78.4
Laplacian+RRSS	68.2	71.8	73.7	74.7	77.2	76.2
SPEC+RRSS	71.0	73.7	74.5	75.3	77.0	76.2
SOGFS+RRSS	71.5	72.2	74.6	76.8	77.0	76.2
Laplacian+ALNR	69.1	71.5	75.0	74.8	75.3	76.5
SPEC+ALNR	72.1	72.6	75.9	75.2	75.0	76.5
SOGFS+ALNR	72.7	74.1	75.8	77.8	76.4	76.5
R-CUR	71.1	73.2	72.7	74.2	76.7	74.5
ALFS-I	76.0	79.2	79.9	79.8	78.6	79.3
ALFS-II	**78.3**	**81.6**	**81.8**	**81.6**	**81.3**	**80.8**

TABLE 6
Accuracy (%) of feature selection + active learning algorithms on the ORL dataset. Best results in each column are highlighted in bold fonts.

(a) SVM

Method	#Dim					
	10	30	50	70	90	512
Laplacian+TED	22.5	52.4	68.5	77.1	81.5	85.6
SPEC+TED	10.5	16.0	19.8	29.2	39.7	85.6
SOGFS+TED	30.3	57.4	72.1	78.2	83.6	85.6
Laplacian+RRSS	36.8	63.2	72.5	78.0	82.2	80.0
SPEC+RRSS	9.3	13.4	16.1	26.1	39.8	80.0
SOGFS+RRSS	40.3	68.9	77.6	82.1	82.3	80.0
Laplacian+ALNR	36.3	62.3	71.4	74.3	78.3	80.2
SPEC+ALNR	9.8	13.8	17.4	26.2	39.2	80.2
SOGFS+ALNR	48.2	72.1	79.1	82.4	83.1	80.2
R-CUR	27.7	64.4	72.4	77.2	82.6	84.7
ALFS-I	55.9	78.3	84.2	86.1	87.7	85.7
ALFS-II	**57.2**	**79.1**	**84.7**	**86.2**	**88.1**	**86.2**

(b) Decision Tree

Method	#Dim					
	10	30	50	70	90	512
Laplacian+TED	12.8	24.9	30.0	33.3	33.5	38.1
SPEC+TED	7.1	11.7	17.3	20.9	25.5	38.1
SOGFS+TED	15.4	27.1	31.5	35.4	36.9	38.1
Laplacian+RRSS	24.7	29.7	32.9	33.7	35.3	33.5
SPEC+RRSS	15.2	18.7	20.8	22.5	25.2	33.5
SOGFS+RRSS	25.1	30.8	33.1	34.1	34.6	33.5
Laplacian+ALNR	24.3	31.3	31.5	32.3	34.1	35.7
SPEC+ALNR	15.0	18.3	20.7	22.2	25.3	35.7
SOGFS+ALNR	26.3	32.5	33.2	34.6	34.7	35.7
R-CUR	18.6	24.4	26.6	29.3	26.1	31.6
ALFS-I	25.5	31.7	35.1	34.0	35.9	38.8
ALFS-II	**29.8**	**35.6**	**37.9**	**37.1**	**37.2**	**40.5**

Coupling of Active Learning and Feature Selection In order to further show the coupling of active learning and feature, i.e., noisy and redundant features can bring adverse effect on sample selection, while 'good' samples will be beneficial to feature selection, we conduct deep studies on the TOX-171 dataset. We first show the benefit to active learning through embedding feature selection. The results are listed in Table 8(a) and Table 8(b). In Table 8(a), when fixing the number of the queries, the performance of using a small subset of all the features is always better than that of using all the features. And for Table 8(b), the accuracies of using a subset of all the features are supervisor to those of using all the features under most of the cases. Even though it is not higher, the performance of using feature subset is still comparable to that of using all the features. Therefore, it is obvious that embedding feature selection is good for learning representative samples.

Next, we will demonstrate that active learning is also helpful to learning informative features. In order to make our experiments more practical and challenging, we add 20% noisy data samples into the original dataset to form a new dataset. The noisy variable is sampled from the standard normal distribution, and the noisy label is drawn from the discrete uniform distribution on $[1, 2, 3, 4]$. Based on the new dataset, we randomly divide it into two parts: one part is used as the candidate set to query representative

TABLE 7
Accuracy (%) of feature selection + active learning algorithms on the FG-NET dataset. Best results in each column are highlighted in bold fonts.

(a) SVM

Method	#Dim					
	10	30	50	70	90	907
Laplacian+TED	37.3	44.8	47.6	53.3	54.6	54.8
SPEC+TED	37.4	37.0	37.0	37.2	37.7	54.8
SOGFS+TED	32.0	38.8	42.6	45.1	47.6	54.8
Laplacian+RRSS	36.8	49.8	52.2	54.5	54.6	51.7
SPEC+RRSS	35.8	35.1	35.1	35.4	35.0	51.7
SOGFS+RRSS	41.1	46.6	49.4	51.7	52.8	51.7
Laplacian+ALNR	37.3	50.3	52.7	55.0	55.1	54.5
SPEC+ALNR	36.2	36.1	36.5	37.0	37.0	54.5
SOGFS+ALNR	41.6	44.2	48.0	50.1	52.4	54.5
R-CUR	41.3	47.1	48.3	50.1	51.1	53.5
ALFS-I	49.3	52.3	55.5	56.0	56.5	54.7
ALFS-II	**49.6**	**53.4**	**55.8**	**56.4**	**57.3**	**55.7**

(b) Decision Tree

Method	#Dim					
	10	30	50	70	90	907
Laplacian+TED	37.7	39.0	38.6	39.7	40.1	41.6
SPEC+TED	33.9	32.4	33.1	35.7	34.2	41.6
SOGFS+TED	32.7	33.1	35.1	37.5	39.2	41.6
Laplacian+RRSS	30.4	39.1	39.6	40.9	41.4	40.3
SPEC+RRSS	32.1	33.0	33.5	33.5	34.5	40.3
SOGFS+RRSS	34.9	37.5	38.5	41.0	39.5	40.3
Laplacian+ALNR	32.8	40.1	40.3	41.2	42.4	41.6
SPEC+ALNR	33.1	34.8	34.7	35.1	34.8	41.6
SOGFS+ALNR	33.2	36.0	37.3	38.9	40.2	41.6
R-CUR	32.9	36.6	38.0	39.2	38.8	41.1
ALFS-I	39.6	41.3	42.1	42.8	43.9	43.7
ALFS-II	**40.3**	**43.2**	**42.8**	**44.1**	**44.8**	**44.4**

TABLE 8
Results (%) showing feature selection being good for learning representative samples on the TOX-171 dataset. Best results in each column are highlighted in bold fonts.

(a) SVM

#Dim	#Query						
	10	20	30	40	50	60	70
10	44.4	51.2	52.8	54.0	56.4	57.7	58.4
20	46.5	53.0	57.4	61.9	62.2	63.4	63.5
30	47.3	56.1	58.7	63.7	65.1	67.2	68.0
40	48.6	56.4	59.8	64.8	67.1	67.3	68.1
50	49.0	57.6	61.7	65.2	67.4	68.5	**69.3**
60	**50.1**	57.7	64.1	66.2	68.4	68.8	68.4
70	48.7	58.4	63.4	66.6	68.0	68.8	69.0
80	48.0	57.8	63.3	66.4	68.0	**69.4**	68.1
90	47.9	57.9	63.8	**67.2**	**69.1**	69.2	69.0
100	47.8	**58.5**	**64.2**	66.4	68.6	69.0	68.6
5748	40.8	46.7	45.2	46.9	43.5	40.6	40.7

(b) Decision Tree

#Dim	#Query						
	10	20	30	40	50	60	70
10	41.5	48.0	50.8	55.0	56.5	57.6	57.8
20	42.6	50.4	54.5	58.0	59.8	60.7	63.0
30	**43.6**	51.1	56.1	58.8	61.3	62.4	62.8
40	**43.6**	52.7	55.8	59.5	61.3	61.7	63.0
50	43.0	51.3	55.9	60.7	62.0	62.1	62.8
60	42.3	52.1	55.5	60.9	61.7	63.1	63.6
70	42.8	52.7	56.7	62.2	62.4	63.4	64.3
80	42.8	52.8	56.9	**62.7**	63.4	64.3	64.5
90	42.8	52.8	56.4	61.7	63.1	64.0	64.9
100	42.8	52.8	57.6	61.7	**64.1**	**64.7**	64.8
5748	43.3	**53.5**	**59.1**	61.4	63.4	64.5	**65.8**

TABLE 9
Results (%) showing active leaning being helpful for selecting informative features on the extended TOX-171 dataset. Best results in each column are highlighted in bold fonts.

(a) SVM

#Query	#Dim					
	10	30	50	70	90	5748
10	41.8	41.8	39.8	41.8	43.7	41.8
20	44.7	49.5	49.5	48.5	48.5	37.9
30	**54.4**	51.5	54.4	54.4	53.4	40.8
40	49.5	54.4	53.4	54.4	55.3	40.8
50	50.5	**57.3**	59.2	59.2	57.3	**46.6**
60	49.5	55.3	58.3	58.3	**58.3**	39.8
70	48.5	**57.3**	**60.2**	**60.2**	**58.3**	42.7
80	50.5	**57.3**	58.3	55.3	**58.3**	41.8
90	46.6	54.4	58.3	57.3	54.4	27.2
100	46.6	53.4	57.3	56.3	53.4	27.2
102	47.6	53.4	58.3	57.3	52.4	24.3

(b) Decision Tree

#Query	#Dim					
	10	30	50	70	90	5748
10	36.9	35.9	37.9	37.9	37.9	37.9
20	43.7	43.7	43.7	44.7	45.6	42.7
30	44.7	49.5	48.5	48.5	50.5	55.3
40	47.6	50.5	54.4	55.3	55.3	54.4
50	53.4	50.5	55.3	**58.3**	58.3	56.3
60	51.5	52.4	52.4	55.3	53.4	58.3
70	**57.3**	**54.4**	**57.3**	55.3	58.3	**59.2**
80	47.6	48.5	50.5	51.5	52.4	**59.2**
90	49.5	53.4	50.5	50.5	53.4	**59.2**
100	51.5	49.5	52.4	56.3	**60.2**	55.3
102	51.5	49.5	51.5	53.4	56.3	56.3

samples for training, and the other part is used as the testing set. When querying the selected samples, we use SVM and decision tree as the final classifier for classifying the testing data, respectively. Table 9(a) and Table 9(b) report the results. With the fixed feature dimensions, querying all the samples, i.e., 102 samples, for training can not obtain the best classification performance. In contrast, only select a small subset of samples for requesting human labeling can significantly improve the classification accuracy, compared with querying all the samples. This indicates that 'good' samples are indeed beneficial to learning informative features.

4.3 CPU Time and Sensitivity Analysis

We test the CPU running time with different convergence tolerance ϵ on the Madelon dataset and the FG-NET dataset. The experiments are conducted on a laptop with Intel(R)-Core(TM) CPUs of 3.20 GHz and 4 GB RAM, and ALFS-II is implemented using MATLAB R2014b 64bit edition without parallel operation. The result is shown in Fig. 4. The CPU time grows linearly with ϵ increasing on both datasets.

We also study the sensitivity of our algorithm to the parameters, α, β, and λ, on the Madelon dataset. In the experiment, we first fix the number of the selected features to 10, and set the number of the selected samples to 1200.

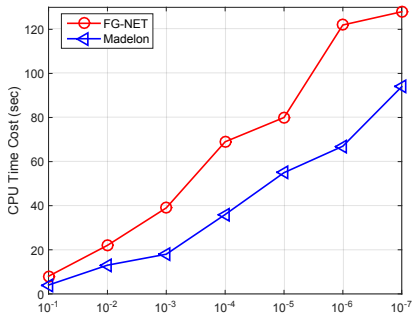
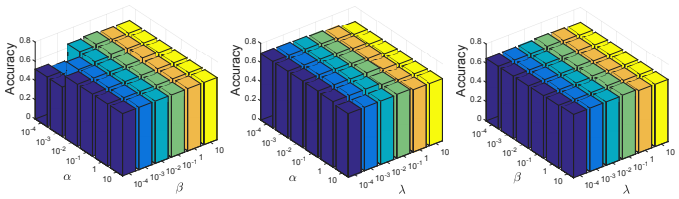


Fig. 4. CPU time vs. convergence tolerance ϵ.



(a) (b) (c)

Fig. 5. Sensitivity study of the parameters on the Madelon dataset.

Then, we fix one parameter, and vary the other two parameters. We report the accuracy of our algorithm with SVM as the final classifier. The results are shown in Fig. 5. We can see that our method is not sensitive to all the parameters with wide ranges. In Fig. 5(a), when α and β are set to small values, the performance of the model degrades significantly. This is because the smaller α and β are, the lower the weight of the second and the third terms in (6) is. In this way, it is hard to guarantee that the columns and the rows of the matrix $\mathbf{W}$ are sparse, which makes our algorithm fail to learn representative samples and features. Therefore, we should set larger α and β in practice.

5 CONCLUSIONS AND FUTURE WORK

In this paper, we present a unified framework to simultaneously conduct active sample learning and feature selection (ALFS). Given an unlabeled dataset, our formulation naturally and effectively incorporates feature and sample selection by solving a regularized optimization problem rooted from CUR factorization. We further relax the original NP-hard non-convex problem into a convex one by introducing the structured sparsity-inducing norms, which allows for efficient iterative optimization algorithm (ADMM). The superior performance of our method over the state-of-the-art methods is verified by extensive experimental evaluations with six benchmark datasets.

Several interesting directions can be followed up, which are not covered by our current work:

- **Leveraging labeled samples**: ALFS selects samples and features from a perspective of data reconstruction in an unsupervised setting. If label information is available, we can incorporate such prior information into our framework, e.g., taking the objective function of [52] as a regularization term. This would be helpful if a specific prediction task is actually only relevant to a few features and our 'blind' feature selection method may keep unnecessary features although they are indispensable to represent the sample set itself.
- **Online learning**: ALFS works in a batch mode, i.e., the unlabeled dataset is available. We can further extend our work to online learning mode, such that ALFS can efficiently and effectively handle the case when new samples are coming in.
- **Additional regularization terms**: In our work, motivated by the local reconstruction philosophy, we add the cross-sample regularization term as presented in Sec.2.3. This term alleviates the underdetermination condition of the factorization problem, and contributes to the robustness of our method. Symmetrically, a cross-feature regularization term can be also applied.

REFERENCES

[1] Y. Freund, H. S. Seung, E. Shamir, and N. Tishby, "Selective sampling using the query by committee algorithm," *Machine Learning*, vol. 28, no. 2-3, pp. 133–168, 1997.
[2] D. A. Cohn, Z. Ghahramani, and M. I. Jordan, "Active learning with statistical models," *Journal of Artificial Intelligence Research*, vol. 4, pp. 129–145, 1996.
[3] S.-J. Huang, R. Jin, and Z.-H. Zhou, "Active learning by querying informative and representative examples," in *Advances in Neural Information Processing Systems*, 2010, pp. 892–900.
[4] L. Li, X. Jin, S. J. Pan, and J.-T. Sun, "Multi-domain active learning for text classification," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2012, pp. 1086–1094.
[5] E. Elhamifar, G. Sapiro, A. Yang, and S. S. Sasrty, "A convex optimization framework for active learning," in *IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2013, pp. 209–216.
[6] X.-Y. Zhang, S. Wang, and X. Yun, "Bidirectional active learning: A two-way exploration into unlabeled and labeled data set," *IEEE Transactions on Neural Network and Learning Systems (TNNLS)*, 2015.
[7] G.-J. Qi, X.-S. Hua, Y. Rui, J. Tang, and H.-J. Zhang, "Two-dimensional active learning for image classification," in *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2008, pp. 1–8.
[8] P. Jain and A. Kapoor, "Active learning for large multi-class problems," in *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 762–769.
[9] A. J. Joshi, F. Porikli, and N. Papanikolopoulos, "Multi-class active learning for image classification," in *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 2372–2379.
[10] S. Vijayanarasimhan, P. Jain, and K. Grauman, "Far-sighted active learning on a budget for image and video recognition," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 3035–3042.
[11] L. Liang and K. Grauman, "Beyond comparing image pairs: Setwise active learning for relative attributes," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*. IEEE, 2014, pp. 208–215.
[12] S. Vijayanarasimhan and K. Grauman, "Large-scale live active learning: Training object detectors with crawled data and crowds," *International Journal of Computer Vision*, vol. 108, no. 1-2, pp. 97–114, 2014.
[13] Z. Wang and J. Ye, "Querying discriminative and representative samples for batch mode active learning," in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2013, pp. 158–166.
[14] D. D. Lewis and W. A. Gale, "A sequential algorithm for training text classifiers," in *Proceedings of the 17th ACM SIGIR International Conference on Research and Development in Information Retrieval*. Springer-Verlag New York, Inc., 1994, pp. 3–12.
[15] Y. Yang, Z. Ma, F. Nie, X. Chang, and A. G. Hauptmann, "Multi-class active learning by uncertainty sampling with diversity maximization," *International Journal of Computer Vision*, vol. 113, no. 2, pp. 113–127, 2014.

[16] N. Roy and A. McCallum, "Toward optimal active learning through monte carlo estimation of error reduction," in *Proceedings of International Conference on Machine Learning*, 2001, pp. 441–448.

[17] F. Nie, H. Wang, H. Huang, and C. Ding, "Early active learning via robust representation and structured sparsity," in *Proceedings of the 23th International Joint Conference on Artificial Intelligence*. AAAI Press, 2013, pp. 1572–1578.

[18] K. Yu, J. Bi, and V. Tresp, "Active learning via transductive experimental design," in *Proceedings of the 23th International Conference on Machine Learning*. ACM, 2006, pp. 1081–1088.

[19] R. Chattopadhyay, Z. Wang, W. Fan, I. Davidson, S. Panchanathan, and J. Ye, "Batch mode active sampling based on marginal probability distribution matching," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2012, pp. 741–749.

[20] D. Cai and X. He, "Manifold adaptive experimental design for text categorization," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 4, pp. 707–719, 2012.

[21] Y. Hu, D. Zhang, Z. Jin, D. Cai, and X. He, "Active learning via neighborhood reconstruction," in *Proceedings of the 23th International Joint Conference on Artificial Intelligence*. AAAI Press, 2013, pp. 1415–1421.

[22] Z. Zhao and H. Liu, "Spectral feature selection for supervised and unsupervised learning," in *Proceedings of the 24th international conference on Machine learning*. ACM, 2007, pp. 1151–1157.

[23] L. Wolf and A. Shashua, "Feature selection for unsupervised and supervised inference: The emergence of sparsity in a weight-based approach," *The Journal of Machine Learning Research*, vol. 6, pp. 1855–1887, 2005.

[24] J. Hemant and X. Xiaowei, "Using active learning with integrated feature selection," *Technical Report UALR06-02*, 2011.

[25] H. Raghavan, O. Madani, and R. Jones, "Active learning with feedback on features and instances," *The Journal of Machine Learning Research*, vol. 7, pp. 1655–1686, 2006.

[26] X. Kong, W. Fan, and P. S. Yu, "Dual active feature and sample selection for graph classification," in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2011, pp. 654–662.

[27] M. Bilgic, "Combining active learning and dynamic dimensionality reduction." in *SIAM International Conference on Data Mining*, 2012, pp. 696–707.

[28] X. He, M. Ji, C. Zhang, and H. Bao, "A variance minimization criterion to feature selection using laplacian regularization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 10, pp. 2013–2025, 2011.

[29] C. Boutsidis and D. P. Woodruff, "Optimal cur matrix decompositions," *arXiv preprint arXiv:1405.7910*, 2014.

[30] M. W. Mahoney and P. Drineas, "Cur matrix decompositions for improved data analysis," *Proceedings of the National Academy of Sciences*, vol. 106, no. 3, pp. 697–702, 2009.

[31] P. Drineas, M. W. Mahoney, and S. Muthukrishnan, "Relative-error cur matrix decompositions," *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 2, pp. 844–881, 2008.

[32] S. Wang and Z. Zhang, "Improving cur matrix decomposition and the nyström approximation via adaptive sampling," *The Journal of Machine Learning Research*, vol. 14, no. 1, pp. 2729–2769, 2013.

[33] D. Gabay and B. Mercier, "A dual algorithm for the solution of nonlinear variational problems via finite element approximation," *Computers & Mathematics with Applications*, vol. 2, no. 1, pp. 17–40, 1976.

[34] Z. Lin, R. Liu, and Z. Su, "Linearized alternating direction method with adaptive penalty for low-rank representation," in *Advances in Neural Information Processing Systems*, 2011, pp. 612–620.

[35] J. Yang, W. Yin, Y. Zhang, and Y. Wang, "A fast algorithm for edge-preserving variational multichannel image restoration," *SIAM Journal on Imaging Sciences*, vol. 2, no. 2, pp. 569–592, 2009.

[36] Z. Lin, M. Chen, L. Wu, and Y. Ma, "The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices," *Technical report, UIUC Technical Report UILU-ENG-09-2215*, 2009.

[37] Y. Zhang and D.-Y. Yeung, "Learning high-order task relationships in multi-task learning," in *Proceedings of the 23-th International Joint Conference on Artificial Intelligence*. AAAI Press, 2013, pp. 1917–1923.

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

[39] B. He, L.-Z. Liao, D. Han, and Y. Hai, "A new inexact alternating directions method for monotone variational inequalities," *Mathematical Programming*, vol. 92, no. 1, pp. 103–118, 2002.

[40] B. He and X. Yuan, "On the o(1/n) convergence rate of the douglas-rachford alternating direction method," *SIAM Journal on Numerical Analysis*, vol. 50, no. 2, pp. 700–709, 2012.

[41] ——, "On non-ergodic convergence rate of douglas–rachford alternating direction method of multipliers," *Numerische Mathematik*, pp. 1–11, 2014.

[42] L. Zhang, C. Chen, J. Bu, D. Cai, X. He, and T. S. Huang, "Active learning based on locally linear reconstruction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 10, pp. 2026–2038, 2011.

[43] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[44] M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml

[45] K. E. Rieger, W.-J. Hong, V. G. Tusher, J. Tang, R. Tibshirani, and G. Chu, "Toxicity from radiation therapy associated with abnormal transcriptional responses to dna damage," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, no. 17, pp. 6635–6640, 2004.

[46] F. Nie, W. Zhu, and X. Li, "Unsupervised feature selection with structured graph optimization," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016, pp. 1302–1308.

[47] H. Han, C. Otto, X. Liu, and A. K. Jain, "Demographic estimation from face images: Human vs. machine performance." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 6, pp. 1148–61, 2015.

[48] C. Li, Q. Liu, J. Liu, and H. Lu, "Learning ordinal discriminative features for age estimation," in *Computer Vision and Pattern Recognition*, 2012, pp. 2570–2577.

[49] ——, "Ordinal distance metric learning for image ranking." *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 7, pp. 1551–1559, 2015.

[50] J. Liu, J. Luo, and M. Shah, "Recognizing realistic actions from videos," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2009, pp. 1996–2003.

[51] X. He, D. Cai, and P. Niyogi, "Laplacian score for feature selection," in *Advances in Neural Information Processing Systems*, 2005, pp. 507–514.

[52] F. Nie, S. Xiang, Y. Jia, C. Zhang, and S. Yan, "Trace ratio criterion for feature selection," in *AAAI'08 Proceedings of the 23rd national conference on Artificial intelligence*, vol. 2, 2008, pp. 671–676.