Unsupervised Active Learning via Subspace Learning

Changsheng Li[1*], Kaihang Mao[1], Lingyan Liang[2], Dongchun Ren[3], Wei Zhang[4], Ye Yuan[1], Guoren Wang[1]

[1]School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China
[2]Inspur Group Company Limited, China, [3]Meituan, Beijing, China
[4]School of Information and Communication Engineering, University of Electronic Science and Technology of China
{lcs,khmao,yuan-ye}@bit.edu.cn, lianglingyan@inspur.com, rendongchun@meituan.com,
zhanggwei1103@163.com, wanggrbit@126.com

Abstract

Unsupervised active learning has been an active research topic in machine learning community, with the purpose of choosing representative samples to be labelled in an unsupervised manner. Previous works usually take the minimization of data reconstruction loss as the criterion to select representative samples, by which the original inputs can be better approximated. However, data are often drawn from low-dimensional subspaces embedded in an arbitrary high-dimensional space in many scenarios, thus it might severely bring in noise if attempting to precisely reconstruct all entries of one observation, leading to a suboptimal solution. In view of this, this paper proposes a novel unsupervised Active Learning model via Subspace Learning, called ALSL. In contrast to previous approaches, ALSL aims to discover low-rank structures of data, and then perform sample selection based on the learnt low-rank representations. To this end, we devise two different strategies and propose two corresponding formulations to select samples with and under low-rank sample representations, respectively. Since the proposed formulations involve several non-smooth regularization terms, we develop a simple but effective optimization procedure to solve them. Extensive experiments are performed on five publicly available datasets, and experimental results demonstrate the proposed first formulation achieves comparable performance with the state-of-the-arts, while the second formulation significantly outperforms them, achieving a 13% improvement over the second best baseline at most.

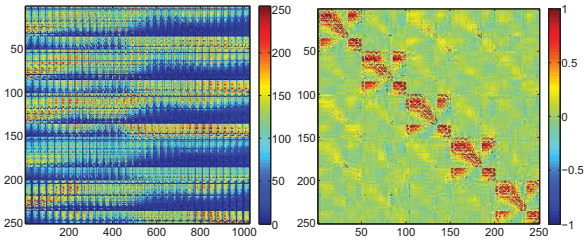
Introduction

Recently, unsupervised active learning has attracted lots of attention in machine learning community. Different from supervised active learning approaches pretraining a classifier by labeled data (Roy and McCallum 2001; Gal, Islam, and Ghahramani 2017; Yoo and Kweon 2019), the goal of unsupervised active learning is to select representative samples to be labelled in an unsupervised manner, so as to better reduce the cost of annotations but still guarantee the performance of downstream models trained by the selected samples.

One key point in unsupervised active learning is how to evaluate representativeness of one sample. To this end, some researchers propose to measure it through minimizing the data reconstruction loss and taking the contributions to reconstructing other samples as its representativeness. Based on this criteria, many unsupervised active learning methods have been proposed in the past decade (Yu, Bi, and Tresp 2006; Zhang et al. 2011; Cai and He 2011; Nie et al. 2013; Hu et al. 2013; Shi and Shen 2016; Li et al. 2019, 2020a,b). Roughly speaking, unsupervised active learning can be divided into two categories: linear and nonlinear ones (Li et al. 2020a). For linear methods, they usually assume that each sample can be well reconstructed by a linear combination of a selected sample subset in the original space. The typical works include transductive experimental design (TED) (Yu, Bi, and Tresp 2006), robust structured representation (RRSS) (Nie et al. 2013), active learning via neighborhood reconstruction (ALNR) (Hu et al. 2013), joint active learning and feature selection (Li et al. 2019). Given the fact that intrinsic structures of data are often complex (e.g., nonlinearity) in practice, a few nonlinear works have been proposed to handle such a case, including the manifold adaptive experimental design (MAED) algorithm (Cai and He 2011) and deep unsupervised active learning (DUAL) (Li et al. 2020a). They usually nonlinearly map original inputs into a latent space, where a linear model is utilized to select samples. In this paper, we focus on linear unsupervised active learning, because of its simplicity but effectiveness.

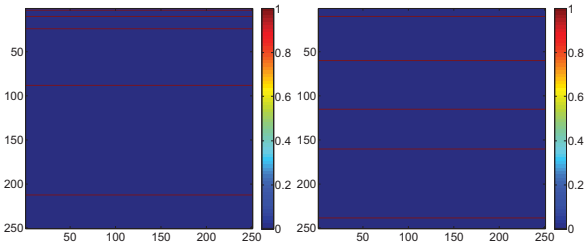
Most of the above linear unsupervised algorithms select representative samples relying on the reconstruction loss minimization in the original space. However, in practice, data are often approximately drawn from low-dimensional subspaces embedded in an arbitrary high-dimensional space. Obviously, it is not necessary to reconstruct all entries of one observation well under such a case. In contrast, if features of the observations are noisy, reconstructing them even degrades model performance, resulting in a suboptimal solution. Thus, it will be beneficial to sample selection, if we can learn and leverage subspace structures of data during training. We take Figure 1 as an example to illustrate the main idea behind our method. Figure 1 (a) is the 2-D exhibition of data matrix from the Extended Yale Face B dataset

*Changsheng Li is the corresponding author.



(a) Original inputs

(b) Learnt feature representations



(c) Selected samples using the original space

(d) Selected samples using the learnt feature representations

Figure 1: Illustration of our basic idea. (a) Each row denotes one sample with an original 1024-dimensional vector. (b) Feature representations learnt by a subspace learning method, LSR (Lu et al. 2012), using the original inputs. (c) The selected samples obtained by a typical active learning method, TED (Yu, Bi, and Tresp 2006), using the original space. (d) An ideal sample selection result obtained by our method using the learnt feature representations. Red lines in Figure 1 (c) and (d) denote the samples are selected.

(Georghiades, Belhumeur, and Kriegman 2001). We use the first 5 subjects, where the first 50 frontal face images for each subject are used in this example. In Figure 1 (a), each row denotes one sample with a 1024-dimensional feature vector. The samples of the first 50 rows are from one subject, i.e., belonging to the same category, and those of the second 50 rows are from the second subject, and so on. From Figure 1 (a), we can observe the first three subjects show similar input patterns, while they are actually drawn from different subspaces as shown in Figure 1 (b). If directly selecting samples in the original space, the result is not good, as shown in Figure 1 (c). In Figure 1 (c), we show five top-ranked samples selected by a typical active learning method, TED (Yu, Bi, and Tresp 2006), where three samples come from the first subject, and there is no sample selected from the third and fourth subjects. If training a classifier based on the selected five samples, the classification performance will be inferior, because of some subjects not selected. An ideal sample selection is demonstrated in Figure 1 (d), obtained through using the learnt representations in Figure 1 (b), rather than the original space in Figure 1 (a). In Figure 1 (d), the five top-ranked samples cover all subjects, which is good for downstream classification tasks.

In light of these, this paper proposes a novel unsupervised Active Learning model based on Subspace Learning, called ALSL. ALSL intends to learn subspace structures of data and perform sample selection in a unified framework. To reach the goal, we devise two different strategies and propose two corresponding formulations to select samples *with* and *under* low-rank sample representations respectively. In the first formulation, we attempt to learn a reconstruction coefficient matrix to minimize the reconstruction loss of original inputs, while simultaneously require the coefficient matrix to be low-rank and row-sparse, such that the samples can be selected with low-rank representations. Different from the first formulation still minimizing the reconstruction loss of original inputs, the proposed second formulation aims to learn low-rank structures of data, and then select representative samples to best approximate the learnt low-rank representations, so as to further suppress noise. Finally, we develop a simple but effective procedure to solve the above two optimization problems. Extensive experiments are performed on multiple tasks usually requiring high annotation costs, and experimental results on five publicly available datasets demonstrate the efficacy of the proposed models.

Related Work

In this section, we review linear unsupervised active learning algorithms which are the most related to our method.

Linear unsupervised active learning intends to utilize a linear model to choose representative samples to be labelled in an unsupervised manner. An earlier approach is the transductive experimental design (TED) (Yu, Bi, and Tresp 2006), the core idea of which is to select a sample subset to best approximate the whole dataset through optimizing a least square loss function plus a ridge regularization term. Later, (Yu et al. 2008) extends TED to a convex formulation by replacing the cardinality constraint by a sparsity regularization. Moreover, (Shi and Shen 2016) further extends TED to simultaneously select representative and diverse samples. Recently, RRSS proposed in (Nie et al. 2013) utilizes a structural sparsity regularization to select samples, and a robust sample representation strategy to mitigate the issue of outliers. However, the complexity of this method is very high when the number of samples is large, which is of order $O(n^4)$. To solve this issue, (FY et al. 2015) proposes an accelerated version to RRSS. ALNR (Hu et al. 2013) aims to incorporate the neighborhood relation of samples into the reconstruction process, so as to make the nearest neighbors of one sample contribute more to the reconstruction of the sample. More recently, a unified framework for simultaneously active learning and feature selection is proposed in (Li et al. 2019), called ALFS, demonstrating that both tasks are beneficial to each other. Most of the above linear methods concentrate on sample selection in the original feature space, ignoring the case that data are often approximately drawn from low-dimensional subspaces embedded into a high-dimensional space.

Proposed Method

Let $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n] \in \mathbb{R}^{d \times n}$ be a collection of data points drawn from a union of multiple low-dimensional subspaces, where d is the dimension of each sample, and n is the

number of samples. Our goal is to perform sample selection and subspace learning simultaneously, making the selected sample subset $\mathbf{S} \in \mathbb{R}^{d\times k}$ more representative. k denotes the number of the selected samples. Before introducing our method, we first summarize the notations and the definition of norms used in this paper.

Vectors are written as boldface lowercase letters and matrices are written as boldface uppercase letters. For an arbitrary matrix $\mathbf{A} \in \mathbb{R}^{n\times m}$, we use $\mathbf{a}^i$, $\mathbf{a}_j$, and a_{ij} to denote its i-th row, j-th column, and (i,j)-th entry, respectively. The Frobenius norm of the matrix $\mathbf{A}$ is defined as $||\mathbf{A}||_F = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{m} a_{ij}^2}$, and the $\ell_{2,1}$-norm of the matrix $\mathbf{A}$ is defined as $||\mathbf{A}||_{2,1} = \sum_{i=1}^{n}\sqrt{\sum_{j=1}^{m} a_{ij}^2} = \sum_{i=1}^{n}||\mathbf{a}^i||_2$. $||\mathbf{A}||_*$ denotes the nuclear norm of $\mathbf{A}$, defined as $||\mathbf{A}||_* = \sum_i \sigma_i$, where σ_i is the i-th singular value of $\mathbf{A}$. The sup-norm of a matrix $\mathbf{A}$ is defined as $||\mathbf{A}||_\infty = \sum_{i=1}^{n}||\mathbf{a}^i||_\infty = \sum_{i=1}^{n}(\max_{1\le j\le m}|a_{ij}|)$. The Euclidean inner product between two matrices is $< \mathbf{A}, \mathbf{B} >= \text{tr}(\mathbf{A}^T\mathbf{B})$, where $\mathbf{A}^T$ is the transpose of a matrix and $\text{tr}(\cdot)$ is the trace of a matrix.

Formulation I: Active Learning *with* Low-Rank Representation

Following previous approaches, we still adopt the strategy of minimizing the overall reconstruction loss in the original space to select the most representative samples. To this end, we take advantage of the following objective function:

$$\min_{\mathbf{Q}\in\mathbb{R}^{n\times n}} ||\mathbf{X} - \mathbf{X}\mathbf{Q}||_F^2 + \eta||\mathbf{Q}||_\ell, \quad (1)$$

where $\eta \ge 0$ is a tradeoff parameter. In (1), the first term aims to minimize the reconstruction loss, where $\mathbf{Q}$ is the reconstruction coefficient matrix. The second terms $||\mathbf{Q}||_\ell$ is a certain norm of $\mathbf{Q}$ used as a regularization term. Since we attempt to select the k most representative samples, the corresponding reconstruction coefficients on these k selected samples should have larger weights, and those of other $n-k$ samples should have as small weights as possible. In an extreme case, if all entries of one row of $\mathbf{Q}$ are zeros, then the corresponding sample will be not selected as an representative one, as it has no any contribution to reconstructing other samples. Thus, $\mathbf{Q}$ should be row-sparse, since the row of matrix $\mathbf{Q}$ reflects the impact of the corresponding sample on reconstructing other samples. To make $\mathbf{Q}$ row-sparse, $||\mathbf{Q}||_\ell$ can be replaced by $||\mathbf{Q}||_{2,1}$ or the sup-norm $||\mathbf{Q}||_\infty$. Here we use $||\mathbf{Q}||_{2,1}$ in this paper. In principle, $||\mathbf{Q}||_\infty$ can also be applied. Actually, the matrix $\mathbf{Q}$ plays two roles in (1): First it is a reconstruction coefficient matrix whose column contains the weights of a linear combination of all samples for reconstructing the corresponding sample; Second it is a self-representation matrix. Each column $\mathbf{q}_i \in \mathbb{R}^n$ in $\mathbf{Q}$ can be taken as a new feature representation of $\mathbf{x}_i$, where $\mathbf{X}$ is regarded as a new dictionary.

As aforementioned, the data are often drawn from low-dimensional subspaces embedded in a high-dimensional space. Thereby, we force $\mathbf{Q}$ to be low-rank, and propose to minimize the following objection function:

$$\min_{\mathbf{Q}\in\mathbb{R}^{n\times n}} ||\mathbf{X} - \mathbf{X}\mathbf{Q}||_F^2 + \eta||\mathbf{Q}||_{2,1} + \lambda\text{rank}(\mathbf{Q}), \quad (2)$$

where $\lambda \ge 0$ is a tradeoff parameter. $\text{rank}(\cdot)$ denotes the rank of a matrix. Minimizing the third term aims to make the learnt $\mathbf{Q}$ low-rank, so as to recover the low-rank structure from the given observation matrix $\mathbf{X}$. However, the above problem is in general NP-hard, a common practice is to replace the rank of $\mathbf{Q}$ by its nuclear norm $||\mathbf{Q}||_*$ (Recht, Fazel, and Parrilo 2010), which leads to a convex formulation as:

$$\min_{\mathbf{Q}\in\mathbb{R}^{n\times n}} ||\mathbf{X} - \mathbf{X}\mathbf{Q}||_F^2 + \eta||\mathbf{Q}||_{2,1} + \lambda||\mathbf{Q}||_*, \quad (3)$$

where $||\mathbf{Q}||_*$ is a convex envelope of the rank function.

Formulation II: Active Learning *under* Low-Rank Representation

In (3), although the proposed formulation can reach the goal of performing unsupervised active learning with low-rank sample representation, but the reconstruction loss is still directly built upon the original input matrix $\mathbf{X}$ for selecting k representative samples. In many cases, the observations often contain many noisy features, due to the low-dimensional subspace structure. If we reconstruct each entry of one observation well, this might severely bring in the noise during training and thus degrade the model performance. As aforementioned, we know that the original inputs can be taken as a dictionary, and the reconstruction coefficients can be regarded as new feature representations of samples. If adding a subspace learning based regularizer on the reconstruction coefficients, e.g., a low-rank constraint, we can obtain the intrinsic low-rank representation of data. Motivated by this point, we propose to jointly learn a low-rank representation, and select k representative samples under the new low-rank representations, instead of the original inputs. To this end, we propose another new formulation as:

$$\begin{aligned}\min_{\mathbf{Q},\mathbf{Z}\in\mathbb{R}^{n\times n}} &||\mathbf{X} - \mathbf{X}\mathbf{Q}||_F^2 + \lambda||\mathbf{Q}||_* \\ &+ \mu||\mathbf{Q} - \mathbf{Q}\mathbf{Z}||_F^2 + \eta||\mathbf{Z}||_{2,1}, \end{aligned} \quad (4)$$

where $\mu \ge 0$ is a tradeoff parameter.

In (4), the first two terms aim to learn a low-rank representation $\mathbf{Q}$, while the last two terms attempt to select the k most representative sample to best approximate the learnt low-rank representation $\mathbf{Q}$. By jointly optimizing these terms, the proposed formulation can perform unsupervised active learning under low-rank representation, making the selected samples more representative. Compared (4) with (3), the main differences are that there is an additional reconstruction loss term in (4) and imposing the $\ell_{2,1}$-norm constraint on different subjects, enabling (4) to perform sample selection under new low-rank representation. Through such changes, the performance of the model can be further improved.

Finally, we utilize $\mathbf{Q}$ to select k most representative samples for the first formulation, and use $\mathbf{Z}$ for the second formulation. To be specific, we can sort all the data points by the l_2-norm of the rows of $\mathbf{Q}$ or $\mathbf{Z}$ in descending order, and select the top k samples as the most representative ones.

Optimization Procedure

The objection function (3) is convex, while it is non-convex in (4). thus (3) and (4) have a globally and locally optimal

solutions respectively. To solve them, we employ the alternating direction method of multipliers (ADMM) (Boyd et al. 2011) to separate the joint problem into easier sub-problems, which could converge to a minimum (Boyd et al. 2011; Hajinezhad et al. 2016) Because of space limitation, we mainly introduce how to optimize (4) using ADMM. The procedure for solving (3) is similar.

In order to solve (4), we first introduce two auxiliary variables $\mathbf{W}$ and $\mathbf{T}$ to convert (4) into the following equivalent objective function:

$$\begin{aligned}\min &||\mathbf{X}-\mathbf{X}\mathbf{Q}||_F^2+\lambda||\mathbf{W}||_*+\mu||\mathbf{Q}-\mathbf{Q}\mathbf{Z}||_F^2+\eta||\mathbf{E}||_{2,1}\\ &s.t.\ \mathbf{Q}=\mathbf{W},\mathbf{Z}=\mathbf{E}.\end{aligned} \tag{5}$$

The augmented Lagrangian function of problem (5) is

$$\begin{aligned}\mathcal{L}_{\rho_1,\rho_2}(\mathbf{Q},\mathbf{Z},\mathbf{W},\mathbf{T},\Lambda_1,\Lambda_2) &:= ||\mathbf{X}-\mathbf{X}\mathbf{Q}||_F^2+\lambda||\mathbf{W}||_*\\ &+\mu||\mathbf{Q}-\mathbf{Q}\mathbf{Z}||_F^2+\eta||\mathbf{E}||_{2,1}+\langle\Lambda_1,\mathbf{Q}-\mathbf{W}\rangle\\ &+\frac{\rho_1}{2}||\mathbf{Q}-\mathbf{W}||_F^2+\langle\Lambda_2,\mathbf{Z}-\mathbf{E}\rangle+\frac{\rho_2}{2}||\mathbf{Z}-\mathbf{E}||_F^2,\end{aligned} \tag{6}$$

where Λ_1 and Λ_2 are Lagrange multipliers. ρ_1 and ρ_2 are constraint violation penalty parameters.

Solver for $\mathbf{W}$: Removing irrelevant terms to $\mathbf{W}$ from (6), it becomes:

$$\min \lambda||\mathbf{W}||_*+\frac{\rho_1}{2}||\mathbf{Q}-\mathbf{W}+\frac{\Lambda_1}{\rho_1}||_F^2. \tag{7}$$

The problem (7) can be solved by the singular value thresholding (Cai, Candes, and Shen 2010), and it has a closed form solution:

$$\mathbf{W}=\mathbf{U}_\mathbf{M}\mathbb{S}_{\frac{\lambda}{\rho_1}}(\Sigma_\mathbf{M})\mathbf{V}_\mathbf{M}^T, \tag{8}$$

where $\mathbb{S}_\tau(x)=sgn(x)\cdot\max(|x|-\tau,0)$ is the soft thresholding operator. $\mathbf{U}_\mathbf{M}\Sigma_\mathbf{M}\mathbf{V}_\mathbf{M}^T$ is the singular value decomposition (SVD) of the matrix $\mathbf{M}$, where $\mathbf{M}=\mathbf{Q}+\frac{\Lambda_1}{\rho_1}$.

Solver for $\mathbf{E}$: $\mathbf{E}$ is the minimizer of

$$\min\ \ \eta||\mathbf{E}||_{2,1}+\frac{\rho_2}{2}||\mathbf{Z}-\mathbf{E}+\frac{\Lambda_2}{\rho_2}||_F^2. \tag{9}$$

The above optimization problem can be solved by the following lemma (Yang et al. 2009):

Lemma 1. *For any $\kappa,\nu>0$, and $\mathbf{g}\in\mathbb{R}^q$, the minimizer of*

$$\min_{\mathbf{t}\in\mathbb{R}^q}\kappa\|\mathbf{t}\|_2+\frac{\nu}{2}\|\mathbf{t}-\mathbf{g}\|_2^2,$$

is given by

$$\mathbf{t}=\begin{cases}(1-\frac{\kappa}{\nu\|\mathbf{g}\|_2})\mathbf{g}, & \|\mathbf{g}\|_2>\frac{\kappa}{\nu}\\ 0, & \|\mathbf{g}\|_2\le\frac{\kappa}{\nu}.\end{cases}$$

Thus, we can obtain the solution of (9) as

$$\mathbf{E}^i=\begin{cases}(1-\frac{\eta}{\rho_2\|\mathbf{v}_i\|_2})\mathbf{v}_i, & \|\mathbf{v}_i\|_2>\frac{\eta}{\rho_2}\\ 0, & \|\mathbf{v}_i\|_2\le\frac{\eta}{\rho_2},\end{cases} \tag{10}$$

where $\mathbf{v}_i=(\mathbf{Z}+\frac{\Lambda_2}{\rho_2})^i$. $\mathbf{E}^i$ denotes the i-th row of $\mathbf{E}$, $i=1,\cdots,n$.

Solver for $\mathbf{Z}$: the sub-problem about $\mathbf{Z}$ can be written as

$$\min\ \ \mu||\mathbf{Q}-\mathbf{Q}\mathbf{Z}||_F^2+\frac{\rho_2}{2}||\mathbf{Z}-\mathbf{E}+\frac{\Lambda_2}{\rho_2}||_F^2. \tag{11}$$

Taking the gradient of (11) with respect to $\mathbf{Z}$, and setting it to zero, we can easily obtain the closed-form solution of $\mathbf{Z}$ as:

$$\mathbf{Z}=(2\mu\mathbf{Q}^T\mathbf{Q}+\rho_2\mathbf{I})^{-1}(2\mu\mathbf{Q}^T\mathbf{Q}+\rho_2\mathbf{E}-\Lambda_2). \tag{12}$$

Solver for $\mathbf{Q}$: when other variables are fixed, $\mathbf{Q}$ can be obtained by minimizing the following objective function:

$$||\mathbf{X}-\mathbf{X}\mathbf{Q}||_F^2+\mu||\mathbf{Q}-\mathbf{Q}\mathbf{Z}||_F^2+\frac{\rho_1}{2}||\mathbf{Q}-\mathbf{W}+\frac{\Lambda_1}{\rho_1}||_F^2. \tag{13}$$

Since (13) is convex in terms of $\mathbf{Q}$, the optimal solution can be found by differentiating (13) and setting the derivative to zero. This implies

$$\begin{aligned}(2\mathbf{X}^T\mathbf{X}+\rho_1\mathbf{I})\mathbf{Q}+2\mu\mathbf{Q}(\mathbf{I}-\mathbf{Z})(\mathbf{I}-\mathbf{Z}^T)&\\ =2\mathbf{X}^T\mathbf{X}+\rho_1\mathbf{W}-\Lambda_1.&\end{aligned} \tag{14}$$

For writing conveniently, let $\mathbf{M}=2\mathbf{X}^T\mathbf{X}+\rho_1\mathbf{I}$, $\mathbf{N}=2\mu(\mathbf{I}-\mathbf{Z})(\mathbf{I}-\mathbf{Z}^T)$, and $\mathbf{H}=2\mathbf{X}^T\mathbf{X}+\rho_1\mathbf{W}-\Lambda_1$, then the Eq. (14) can be written as

$$\mathbf{M}\mathbf{Q}+\mathbf{Q}\mathbf{N}=\mathbf{H}. \tag{15}$$

Since $\mathbf{M}$ and $\mathbf{N}$ are symmetric and positive semi-definite, we have

$$\begin{cases}\mathbf{M}=\mathbf{U}\Sigma_1\mathbf{U}^T\\ \mathbf{N}=\mathbf{V}\Sigma_2\mathbf{V}^T,\end{cases} \tag{16}$$

where $\mathbf{U}$ and $\mathbf{V}$ are both orthogonal. Σ_1 and Σ_2 are two diagonal matrices.

Plugging (16) into (15), we can obtain

$$\begin{aligned}&\mathbf{U}\Sigma_1\mathbf{U}^T\mathbf{Q}+\mathbf{Q}\mathbf{V}\Sigma_2\mathbf{V}^T=\mathbf{H}\\ \Rightarrow&\Sigma_1\mathbf{U}^T\mathbf{Q}+\mathbf{U}^T\mathbf{Q}\mathbf{V}\Sigma_2\mathbf{V}^T=\mathbf{U}^T\mathbf{H}\\ \Rightarrow&\Sigma_1\mathbf{U}^T\mathbf{Q}\mathbf{V}+\mathbf{U}^T\mathbf{Q}\mathbf{V}\Sigma_2=\mathbf{U}^T\mathbf{H}\mathbf{V}.\end{aligned} \tag{17}$$

Let $\mathbf{S}=\mathbf{U}^T\mathbf{Q}\mathbf{V}$, then (17) can be rewritten as

$$\begin{aligned}&\Sigma_1\mathbf{S}+\mathbf{S}\Sigma_2=\mathbf{U}^T\mathbf{H}\mathbf{V}\\ \Rightarrow&\mathbf{S}_{ij}=\frac{(\mathbf{U}^T\mathbf{H}\mathbf{V})_{ij}}{(\Sigma_1)_{ii}+(\Sigma_2)_{jj}},i,j=1,\cdots,n.\end{aligned} \tag{18}$$

Actually, we have $(\Sigma_1)_{ii}>0$ and $(\Sigma_2)_{jj}\ge0$, thus the denominator in (18) is greater than zero. After obtaining $\mathbf{S}$, we can obtain $\mathbf{Q}$ by:

$$\mathbf{Q}=\mathbf{U}\mathbf{S}\mathbf{V}^T. \tag{19}$$

The updating rule for Λ_1,Λ_2: The Lagrange multipliers Λ_1,Λ_2 can be updated by:

$$\begin{cases}\Lambda_1\leftarrow\Lambda_1+\rho_1(\mathbf{Q}-\mathbf{W})\\ \Lambda_2\leftarrow\Lambda_2+\rho_2(\mathbf{Z}-\mathbf{E}).\end{cases} \tag{20}$$

We list the key steps for solving (4) in Algorithm 1. The procedure for solving (3) is similar to that in Algorithm 1.

Algorithm 1: Optimization Procedure for Solving Formulation II

Input: The matrix $\mathbf{X} \in \mathbb{R}^{d\times n}$, tradeoff parameters;
Initialization: $\mathbf{Q}^0 = \mathbf{W}^0 = \mathbf{Z}^0 = \mathbf{E}^0 = 0$, $\Lambda_1^0 = \Lambda_2^0 = 0$, $\rho_1 = \rho_2 = 10^{-6}$, $\max_\rho = 10^6$, $\tau = 1.1$, $\epsilon = 10^{-3}$, $t = 0$;
while *not converged* **do**
1. Fix the other variables and update $\mathbf{W}$ by (8);
2. Fix the other variables and update $\mathbf{E}$ by (10);
3. Fix the other variables and update $\mathbf{Z}$ by (12);
4. Fix the other variables and update $\mathbf{Q}$ by (19);
5. Update the multiplier Λ_1 and Λ_2 by (20);
6. Update the parameter ρ_1, ρ_2 by $\rho_i = \min(\tau\rho_i, \max_\rho), i = 1, 2$;
7. Check the convergence conditions: $\|\mathbf{Q} - \mathbf{W}\|_\infty < \epsilon$, $\|\mathbf{Z} - \mathbf{E}\|_\infty < \epsilon$, $|\frac{\mathcal{L}^{t+1} - \mathcal{L}^t}{\mathcal{L}^t}| < \epsilon$, where $\mathcal{L}^t$ is the value of the augmented Lagrangian function (6) for the t-th iteration;
8. $t \leftarrow t + 1$;

end
Output: $\mathbf{Z}$.

Convergence and Computational Complexity Analysis

It is easy to proof that Algorithm 1 will converge to a local optimum based on (Hajinezhad et al. 2016). Because of space limitation, we omit the proof. Please refer to (Hajinezhad et al. 2016) for the details.

Moreover, we analyze the computational complexities of Algorithm 1. The main computation costs in Algorithm 1 stem from updating $\mathbf{W}, \mathbf{E}, \mathbf{Z}$, and $\mathbf{Q}$. For updating $\mathbf{W}$, it typical needs $O(n^3)$ because of referring to SVD of an $n \times n$ matrix. The complexity is of order $O(n^2)$ for updating $\mathbf{E}$. It costs $O(n^3)$ for updating $\mathbf{Z}$, since it needs several matrix multiplications and an inverse operation of a matrix. As for $\mathbf{Q}$, it requires $O(n^3 + n^2 d)$. Thus, the total complexity of Algorithm 1 in each iteration is $O(n^3{+}n^2{+}n^3{+}n^3{+}n^2 d) = O(n^3{+}n^2 d)$. We know ALFS is of order $O(n^3{+}n^2 d{+}d^2 n{+}d^3)$ and RRSS needs $O(n^4)$ in each iteration, where n is the number of samples and d is the dimension of samples. Thus, our method has lower complexity than ALFS and RRSS.

Experiment

In this section, we evaluate the performance of the proposed formulations on five publicly available datasets across different tasks including facial age estimation, video action recognition, medical image classification and wine quality prediction. It is usually much more time-consuming and expensive to manually label samples for these tasks.

Datasets

We evaluate the proposed methods on two video action recognition datasets HMDB51 (Kuehne et al. 2011) and UCF50 (Reddy and Shah 2013), one facial age estimation datasets UTKFace (Zhang, Song, and Qi 2017), one medical image dataset HAM10000 (Tschandl, Rosendahl, and Kittler 2018), and one wine quality dataset (Cortez et al. 2009). HMDB51 is composed of 6,766 realistic video clips from 51 action categories. UCF50 contains 6,681 realistic youtube videos associated with 50 action categories. For each video, we extract a 512-dimensional feature vector to represent each frame, and then use the average of all frames as the feature representation of the video. UTKFace consists of over 20,000 face images with annotations of age. The age spans range from 0 to 116 years old, which are divided into eight ranges: 0-9, 10-19, ..., 60-69, and 70+ for age group estimation. We randomly select 200 images from each class, in order to avoid the effect of class imbalance.

Compared Methods and Experimental Protocol

We compare with some related unsupervised active learning algorithms, including RRSS (Nie et al. 2013), ALNR (Hu et al. 2013), ALFS (Li et al. 2019). In addition, we also compare with a popular matrix column subset selection algorithm, DCS (Papailiopoulos, Kyrillidis, and Boutsidis 2014) which can be used for sample selection. There are two variants about our method, ALSL_W and ALSL_U, meaning that performing sample selection with and under low-rank sample representations, respectively. To evaluate the effectiveness of the proposed method, we train a SVM classifier using the selected samples, and evaluate its accuracy using the unseen samples. The parameters λ, μ and η in our algorithm are searched from $\{0.001, 0.01, 0.1, 1, 10\}$. In the experiment, we repeat every test case 5 times, and report the average result and standard deviation.

Experimental Results

General Performance: The results are listed in Table 1, 2, 3, 4, and 5. ALSL_U consistently outperforms all other methods on the five datasets. For instance, when setting the number of queries to 100 on the UCF50 dataset, ALSL_U obtains a 13% improvement over DCS that achieves the second best performance. This illustrates the strategy for selecting samples to best approximate the low-rank representations is effective. ALSL_W has a comparable performance with other baselines. This is because ALSL_W aims to reconstruct the original inputs, leading to struggling with noise.

Ablation Study: We study the effectiveness of our components on the two image datasets. The experimental setting is as follows: We only perform sample selection with subspace learning, i.e., setting $\lambda = 0$ in (3), shorten as AL. The experimental results are reported in Table 6. ALSL_W is better than AL, indicating that learning low-rank sample representations is good for sample selection. ALSL_U achieves superior performance over ALSL_W, which demonstrates that selecting samples to reconstruct the low-rank representations can better suppress noise.

Effectiveness of Joint Learning: In (4), we propose a joint framework to perform subspace learning and sample selection simultaneously. Someone may argue that whether we can first learn a low-rank representation $\mathbf{Q}$ by the first second terms of (4), and then select samples to best approximate

#Num.	ALFS	ALNR	DCS	RRSS	ALSL_W	ALSL_U
50	0.184±0.007	0.131±0.016	0.127±0.011	0.150±0.018	0.218±0.024	**0.242**±0.018
100	0.196±0.024	0.160±0.018	0.133±0.015	0.214±0.018	0.243±0.010	**0.273**±0.015
150	0.259±0.011	0.183±0.014	0.153±0.015	0.233±0.009	0.285±0.035	**0.301**±0.022
200	0.299±0.024	0.190±0.011	0.221±0.018	0.275±0.017	0.303±0.018	**0.317**±0.003
250	0.318±0.019	0.213±0.014	0.260±0.015	0.302±0.015	0.326±0.013	**0.351**±0.023
300	0.331±0.024	0.200±0.013	0.295±0.011	0.317±0.019	0.340±0.026	**0.374**±0.004
350	0.351±0.014	0.254±0.022	0.315±0.017	0.352±0.011	0.356±0.028	**0.380**±0.010
400	0.355±0.024	0.265±0.012	0.328±0.021	0.349±0.024	0.368±0.017	**0.392**±0.004
450	0.371±0.015	0.297±0.012	0.355±0.027	0.379±0.009	0.376±0.013	**0.398**±0.014

Table 1: Quantitative results in terms of accuracy of different active learning methods on the UTKFace dataset.

#Num.	ALFS	ALNR	DCS	RRSS	ALSL_W	ALSL_U
50	0.227±0.015	0.204±0.020	0.126±0.013	0.191±0.018	0.256±0.019	**0.271**±0.020
100	0.233±0.014	0.232±0.015	0.127±0.010	0.227±0.011	0.276±0.018	**0.295**±0.015
150	0.274±0.020	0.253±0.022	0.206±0.020	0.239±0.014	0.285±0.019	**0.328**±0.020
200	0.302±0.012	0.284±0.010	0.259±0.010	0.280±0.019	0.295±0.019	**0.331**±0.013
250	0.344±0.016	0.301±0.023	0.268±0.015	0.287±0.013	0.302±0.011	**0.355**±0.018
300	0.359±0.011	0.312±0.015	0.311±0.010	0.319±0.010	0.318±0.020	**0.374**±0.017
350	0.368±0.021	0.335±0.020	0.339±0.014	0.326±0.014	0.334±0.015	**0.399**±0.019
400	0.379±0.017	0.358±0.010	0.357±0.010	0.346±0.015	0.351±0.011	**0.402**±0.019
450	0.393±0.018	0.352±0.019	0.359±0.014	0.369±0.019	0.360±0.010	**0.407**±0.020

Table 2: Quantitative results in terms of accuracy of different active learning methods on the HAM10000 dataset.

#Num.	ALFS	ALNR	DCS	RRSS	ALSL_W	ALSL_U
50	0.117±0.010	0.111±0.018	0.115±0.017	0.122±0.015	0.108±0.020	**0.159**±0.011
100	0.154±0.017	0.143±0.011	0.182±0.028	0.157±0.021	0.159±0.020	**0.207**±0.012
150	0.208±0.009	0.180±0.006	0.214±0.014	0.208±0.010	0.207±0.010	**0.243**±0.010
200	0.220±0.016	0.208±0.013	0.254±0.007	0.226±0.023	0.234±0.022	**0.269**±0.012
250	0.266±0.012	0.247±0.027	0.277±0.014	0.281±0.015	0.257±0.025	**0.294**±0.007
300	0.279±0.020	0.276±0.016	0.302±0.015	0.290±0.019	0.277±0.014	**0.318**±0.008
350	0.314±0.012	0.300±0.010	0.324±0.012	0.327±0.014	0.302±0.022	**0.344**±0.007
400	0.326±0.018	0.313±0.014	0.341±0.016	0.330±0.010	0.323±0.013	**0.368**±0.003
450	0.352±0.009	0.333±0.009	0.367±0.009	0.364±0.009	0.339±0.008	**0.377**±0.009

Table 3: Quantitative results in terms of accuracy of different active learning methods on the HMDB51 dataset.

#Num.	ALFS	ALNR	DCS	RRSS	ALSL_W	ALSL_U
50	0.142±0.016	0.152±0.020	0.186±0.017	0.185±0.018	0.130±0.008	**0.249**±0.010
100	0.237±0.017	0.242±0.022	0.246±0.017	0.240±0.010	0.246±0.003	**0.377**±0.011
150	0.340±0.015	0.310±0.014	0.297±0.011	0.345±0.014	0.335±0.010	**0.450**±0.010
200	0.389±0.019	0.370±0.024	0.404±0.021	0.374±0.011	0.406±0.006	**0.511**±0.008
250	0.476±0.010	0.422±0.012	0.449±0.012	0.441±0.013	0.450±0.018	**0.565**±0.010
300	0.525±0.015	0.468±0.012	0.482±0.017	0.469±0.023	0.488±0.008	**0.607**±0.005
350	0.564±0.011	0.510±0.015	0.522±0.012	0.495±0.013	0.512±0.012	**0.641**±0.017
400	0.597±0.013	0.542±0.017	0.551±0.009	0.506±0.023	0.555±0.007	**0.667**±0.009
450	0.628±0.008	0.581±0.017	0.580±0.006	0.532±0.013	0.598±0.007	**0.684**±0.014

Table 4: Quantitative results in terms of accuracy of different active learning methods on the UCF50 dataset.

#Num.	ALFS	ALNR	DCS	RRSS	ALSL_W	ALSL_U
50	0.447±0.007	0.428±0.009	0.448±0.024	0.472±0.002	0.451±0.012	**0.474**±0.007
100	0.441±0.018	0.443±0.011	0.466±0.010	0.453±0.010	0.417±0.023	**0.498**±0.009
150	0.471±0.017	0.479±0.005	0.477±0.010	0.475±0.024	0.490±0.008	**0.507**±0.015
200	0.475±0.011	0.479±0.024	0.494±0.017	0.481±0.024	0.499±0.013	**0.510**±0.007
250	0.502±0.017	0.490±0.018	0.503±0.009	0.495±0.013	0.501±0.011	**0.516**±0.006
300	0.505±0.017	0.498±0.013	0.504±0.026	0.496±0.014	0.500±0.015	**0.521**±0.006
350	0.509±0.018	0.504±0.017	0.515±0.014	0.503±0.023	0.509±0.013	**0.526**±0.006
400	0.511±0.017	0.508±0.017	0.517±0.019	0.503±0.013	0.511±0.011	**0.528**±0.001
450	0.517±0.013	0.511±0.024	0.519±0.016	0.512±0.023	0.519±0.019	**0.534**±0.006

Table 5: Quantitative results in terms of accuracy of different active learning methods on the Wine Quality dataset.

(a) UTKFace

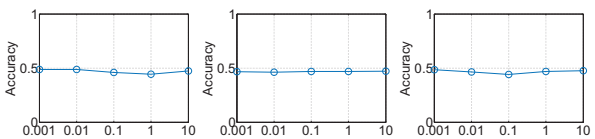
# Num.	AL	ALSL_W	ALSL_U
50	0.130±0.013	0.218±0.024	**0.242**±0.018
100	0.223±0.016	0.243±0.010	**0.273**±0.015
150	0.237±0.026	0.285±0.035	**0.301**±0.022
200	0.264±0.027	0.303±0.018	**0.317**±0.003
250	0.265±0.013	0.326±0.013	**0.351**±0.023
300	0.293±0.018	0.340±0.026	**0.374**±0.004
350	0.322±0.016	0.356±0.028	**0.380**±0.010
400	0.331±0.016	0.368±0.017	**0.392**±0.004
450	0.335±0.014	0.376±0.013	**0.398**±0.014

(b) HAM10000

# Num.	AL	ALSL_W	ALSL_U
50	0.156±0.036	0.256±0.019	**0.271**±0.020
100	0.196±0.009	0.276±0.018	**0.295**±0.015
150	0.201±0.010	0.285±0.019	**0.328**±0.020
200	0.207±0.032	0.295±0.019	**0.331**±0.013
250	0.254±0.019	0.302±0.011	**0.355**±0.018
300	0.272±0.021	0.318±0.020	**0.374**±0.017
350	0.271±0.016	0.334±0.015	**0.399**±0.019
400	0.290±0.023	0.351±0.011	**0.402**±0.019
450	0.321±0.022	0.360±0.010	**0.407**±0.019

Table 6: Ablation study of the proposed methods.

Q using the last two terms in (4). To verify the effectiveness of joint learning, we compare ALSL_U with the results using two separate steps. The results are listed in Table 7. It is obvious that ALSL_U achieves better performance, stemming from the coupling effect of active learning and subspace learning, as verified in (Li et al. 2019).



(a) λ (b) μ (c) η

Figure 2: Parameter Study on the Wine Quality dataset.

Parameter Study: We also study the parameter sensitivities of our method on the Wine Quality dataset. In the experiment, the number of the selected samples is set to 50. The results of ALSL_W are shown in Figure 2. Our method is not sensitive to the three parameters with wide ranges.

(a) HMDB51

#Num.	Separate Steps	ALSL_U
50	0.101±0.022	**0.159**±0.011
100	0.168±0.020	**0.207**±0.012
150	0.204±0.018	**0.243**±0.010
200	0.238±0.016	**0.269**±0.012
250	0.251±0.016	**0.294**±0.007
300	0.282±0.016	**0.318**±0.008
350	0.305±0.012	**0.344**±0.007
400	0.316±0.021	**0.368**±0.003
450	0.326±0.020	**0.377**±0.009

(b) UCF50

#Num.	Separate Steps	ALSL_U
50	0.107±0.022	**0.249**±0.010
100	0.166±0.020	**0.377**±0.011
150	0.234±0.018	**0.450**±0.010
200	0.299±0.016	**0.511**±0.008
250	0.357±0.016	**0.565**±0.010
300	0.395±0.016	**0.607**±0.005
350	0.441±0.012	**0.641**±0.017
400	0.480±0.021	**0.667**±0.009
450	0.523±0.020	**0.684**±0.014

Table 7: Effectiveness verification of joint learning on the HMDB51 and UCF50 datasets.

Conclusion

In this paper, we proposed a subspace learning based model for unsupervised active learning. To leverage subspace learning, we proposed two formulations to perform sample selection *with* and *under* low-rank sample representations respectively. Experimental results on multiple tasks demonstrated their effectiveness. Moreover, we can come to the conclusion that selecting samples by approximating low-rank representations of data can obtain best performance.

Acknowledgments

This work was supported in part by the NSFC under Grants 61806044, U2001211, 61932004, and 61732003. It also was partially supported by the Open Project Program of the National Laboratory of Pattern Recognition (NLPR).

References

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

Cai, D.; and He, X. 2011. Manifold adaptive experimental design for text categorization. *TKDE* 24(4): 707–719.

Cai, J.; Candes, E. J.; and Shen, Z. 2010. A Singular Value Thresholding Algorithm for Matrix Completion. *Siam Journal on Optimization* 20(4): 1956–1982.

Cortez, P.; Cerdeira, A.; Almeida, F.; Matos, T.; and Reis, J. 2009. Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems* 47(4): 547–553.

FY, Z.; Zhu, X.; SM, X.; CH, P.; et al. 2015. 10,000+ Times Accelerated Robust Subset Selection (ARSS). In *Proceedings of The AAAI Conference on Artificial Intelligence (AAAI)*, 3217–3223.

Gal, Y.; Islam, R.; and Ghahramani, Z. 2017. Deep bayesian active learning with image data. In *Proceedings of International Conference on Machine Learning (ICML)*, 1183–1192.

Georghiades, A. S.; Belhumeur, P. N.; and Kriegman, D. J. 2001. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 23(6): 643–660.

Hajinezhad, D.; Chang, T.-H.; Wang, X.; Shi, Q.; and Hong, M. 2016. Nonnegative matrix factorization using admm: Algorithm and convergence analysis. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 4742–4746. IEEE.

Hu, Y.; Zhang, D.; Jin, Z.; Cai, D.; and He, X. 2013. Active learning via neighborhood reconstruction. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI)*, 1415–1421.

Kuehne, H.; Jhuang, H.; Garrote, E.; Poggio, T.; and Serre, T. 2011. HMDB: a large video database for human motion recognition. In *ICCV*, 2556–2563.

Li, C.; Ma, H.; Kang, Z.; Yuan, Y.; Zhang, X.-Y.; and Wang, G. 2020a. On deep unsupervised active learning. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI)*.

Li, C.; Wang, X.; Dong, W.; Yan, J.; Liu, Q.; and Zha, H. 2019. Joint active learning with feature selection via cur matrix decomposition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 41(6): 1382–1396.

Li, C.; Yang, C.; Liang, L.; Yuan, Y.; and Wang, G. 2020b. On Robust Grouping Active Learning. *IEEE Transactions on Emerging Topics in Computational Intelligence* .

Lu, C.; Min, H.; Zhao, Z.-Q.; Zhu, L.; Huang, D.-S.; and Yan, S. 2012. Robust and efficient subspace segmentation via least squares regression. In *ECCV*, 347–360.

Nie, F.; Wang, H.; Huang, H.; and Ding, C. 2013. Early active learning via robust representation and structured sparsity. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI)*, 1572–1578.

Papailiopoulos, D.; Kyrillidis, A.; and Boutsidis, C. 2014. Provable deterministic leverage score sampling. *ArXiv* abs/1404.1530.

Recht, B.; Fazel, M.; and Parrilo, P. A. 2010. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM review* 52(3): 471–501.

Reddy, K. K.; and Shah, M. 2013. Recognizing 50 human action categories of web videos. *Machine vision and applications* 24(5): 971–981.

Roy, N.; and McCallum, A. 2001. Toward optimal active learning through monte carlo estimation of error reduction. In *Proceedings of International Conference on Machine Learning (ICML)*, 441–448.

Shi, L.; and Shen, Y.-D. 2016. Diversifying convex transductive experimental design for active learning. In *Proceedings of International Joint Conferences on Artificial Intelligence (IJCAI)*, 1997–2003. AAAI Press.

Tschandl, P.; Rosendahl, C.; and Kittler, H. 2018. The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific data* 5: 180161.

Yang, J.; Yin, W.; Zhang, Y.; and Wang, Y. 2009. A fast algorithm for edge-preserving variational multichannel image restoration. *SIAM Journal on Imaging Sciences* 2(2): 569–592.

Yoo, D.; and Kweon, I. S. 2019. Learning Loss for Active Learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 93–102.

Yu, K.; Bi, J.; and Tresp, V. 2006. Active learning via transductive experimental design. In *Proceedings of International Conference on Machine Learning (ICML)*, 1081–1088.

Yu, K.; Zhu, S.; Xu, W.; and Gong, Y. 2008. Non-greedy active learning for text categorization using convex ansductive experimental design. In *Proceedings of International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 635–642.

Zhang, L.; Chen, C.; Bu, J.; Cai, D.; He, X.; and Huang, T. S. 2011. Active Learning Based on Locally Linear Reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 33(10): 2026–2038.

Zhang, Z.; Song, Y.; and Qi, H. 2017. Age Progression/Regression by Conditional Adversarial Autoencoder. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.