On Deep Unsupervised Active Learning

Changsheng Li[1] , **Handong Ma**[2] , **Zhao Kang**[2] , **Ye Yuan**[1] , **Xiao-Yu Zhang**[3] and **Guoren Wang**[1*]
[1]School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China
[2]SCSE, University of Electronic Science and Technology of China, Chengdu, China
[3]Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
{lcs, yuan-ye}@bit.edu.cn, 201921080133@std.uestc.edu.cn, zkang@uestc.edu.cn,
zhangxiaoyu@iie.ac.cn, wanggrbit@126.com

Abstract

Unsupervised active learning has attracted increasing attention in recent years, where its goal is to select representative samples in an unsupervised setting for human annotating. Most existing works are based on shallow linear models by assuming that each sample can be well approximated by the span (i.e., the set of all linear combinations) of certain selected samples, and then take these selected samples as representative ones to label. However, in practice, the data do not necessarily conform to linear models, and how to model nonlinearity of data often becomes the key point to success. In this paper, we present a novel Deep neural network framework for Unsupervised Active Learning, called DUAL. DUAL can explicitly learn a nonlinear embedding to map each input into a latent space through an encoder-decoder architecture, and introduce a selection block to select representative samples in the the learnt latent space. In the selection block, DUAL considers to simultaneously preserve the whole input patterns as well as the cluster structure of data. Extensive experiments are performed on six publicly available datasets, and experimental results clearly demonstrate the efficacy of our method, compared with state-of-the-arts.

1 Introduction

In many real-world applications, there are lots of available unlabeled data whereas labeled data are often difficult to get. It is expensive and time-consuming to manually annotate the data, especially when domain experts must be involved. In this situation, active learning provides a promising way to reduce the cost by automatically selecting the most informative or representative samples from an unlabeled data pool for human labeling. In other words, these selected samples can improve the performance of the model (e.g., classifier) the most if they are labeled and used as training data. Due to its huge potential, active learning has been successfully applied to various tasks such as image classification [Joshi *et al.*, 2009], object detection [Vijayanarasimhan and Grauman, 2014], video recommendation [Cai *et al.*, 2019], etc.

Currently, research works on active learning follow two lines according to whether supervised information is involved [Li *et al.*, 2019]. The first line concentrates on how to leverage data structures to select representative samples in an unsupervised manner. Typical algorithms include transductive experimental design (TED) [Yu *et al.*, 2006], locally linear reconstruction [Zhang *et al.*, 2011], robust representation and structured sparsity [Nie *et al.*, 2013], joint active learning and feature selection (ALFS) [Li *et al.*, 2019]. The other line considers the problems of querying informative samples. Such methods basically need a pre-trained classifier to select samples, which means that they need some initially labeled data for training. In this line, many approaches have been proposed in the past decades [Freund *et al.*, 1997; Kapoor *et al.*, 2007; Jain and Kapoor, 2009; Huang *et al.*, 2010; Elhamifar *et al.*, 2013; Zheng and Ye, 2015; Zhang *et al.*, 2017; Haussmann *et al.*, 2019]. Due to the limitation of space, we refer the reader to [Aggarwal *et al.*, 2014] and [Settles, 2009] for more details. In this paper, we focus on unsupervised active learning, since it is a challenging problem because of the lack of supervised information.

Most existing works on unsupervised active learning [Yu *et al.*, 2006; Nie *et al.*, 2013; Hu *et al.*, 2013; FY *et al.*, 2015; Shi and Shen, 2016; Li *et al.*, 2017; 2019] assume that each data point can be reconstructed by the span, i.e., the set of all linear combinations, of a selected sample subset, and resort to shallow linear models to minimize the reconstruction error. Such methods often suffer from the following limitations: first, they attempt to reconstruct the whole dataset, but ignore the cluster structure of data. For example, let us consider an extreme case: assume that there are 100 samples, among which 99 samples are positive and one sample is negative. Thus, the negative sample is very important, and it should be selected as one of the representative samples. However, if we only minimize the total reconstruction loss of all samples by a selected sample subset, then it is very likely that the negative sample is not selected, because of it being far from the positive samples. The second limitation of these methods is that they use shallow and linear mapping functions to reveal the intrinsic structure of data, thus may fail in handling data with complex (often nonlinear) structures. To address this issue, the manifold adaptive experimental design (MAED) algorith-

[*] Guoren Wang is the corresponding author.

m [Cai and He, 2011] attempts to learn the representation in a manifold adaptive kernel space obtained by incorporating the manifold structure into the reproducing kernel Hilbert space (RKHS). As we know, kernel based methods heavily depend on the choice of kernel functions, while it is difficult to find an optimal or even suitable kernel function in real-world applications.

To overcome the above limitations, in this paper, we propose a novel unsupervised active learning framework based on a deep learning model, called Deep Unsupervised Active Learning (DUAL). DUAL takes advantage of an encoder-decoder model to explicitly learn a nonlinear latent space, where DUAL can perform sample selection by introducing a *selection* block at the junction between the encoder and the decoder. In the selection block, we attempt to simultaneously reconstruct the whole dataset and the cluster centroids of the data. Since the selection block is differentiable, DUAL is an end-to-end trainable framework. To the best of our knowledge, our approach constitutes the first attempt to select representative samples based on a deep neural network in an unsupervised setting. Compared with existing unsupervised active learning approaches, our method significantly differs from them in the following aspects:

- DUAL directly learns a nonlinear representation by a multi-layer neural network, which offers stronger ability to discover nonlinear structures of data.
- In contrast to kernel-based approaches, DUAL can provide explicit transformations, avoiding subjectively choosing the kernel function in advance.
- Not only can DUAL model the whole input patterns by the selected representative samples, but also it can preserve cluster structures of data well.

We extensively evaluate our method on six publicly available datasets. The experimental results show our DUAL outperforms the related state-of-the-art methods.

2 Related Work

As mentioned above, we focus on the studies on unsupervised active learning. In this section, we briefly review some algorithms devoted to unsupervised active learning and some related works to this topic.

2.1 Unsupervised Active Learning

Among existing unsupervised active learning methods, the most representative one is the transductive experimental design (TED) [Yu *et al.*, 2006], where its goal is to select samples that can best represent the dataset using a linear representation. Thus, TED proposes an objective function as:

$$\min_{\mathbf{Z},\mathbf{S}} \sum_{i=1}^{n} (||\mathbf{x}_i - \mathbf{Z}\mathbf{s}_i||_2^2 + \alpha||\mathbf{s}_i||_2^2)$$
$$s.t.\ \mathbf{S} = [\mathbf{s}_1, ..., \mathbf{s}_n] \in \mathbb{R}^{m\times n}, \mathbf{Z} = [\mathbf{z}_1, ..., \mathbf{z}_m] \subset \mathbf{X}, \quad (1)$$

where $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_n] \in \mathbb{R}^{d\times n}$ denotes the data matrix, where d is the dimension of samples and n is the number of samples. $\mathbf{Z} \subset \mathbf{X}$ denotes the selected sample subset, and $\mathbf{S}$ is the reconstruction coefficient matrix. α is a tradeoff parameter to control the amount of shrinkage. $||\cdot||_2$ denotes the l_2-norm of a vector.

Following TED, more unsupervised active learning methods have been proposed in recent years. Inspired by the idea of Locally Linear Embedding (LLE) [Roweis and Saul, 2000], [Zhang *et al.*, 2011] propose to represent each sample by a linear combination of its neighbors, with the purpose of preserving the intrinsic local structure of data. Similarly, ALNR also incorporates the neighborhood relation into the sample selection process, where the nearest neighbors are expected to have stronger effect on the reconstruction of samples [Hu *et al.*, 2013]. [Nie *et al.*, 2013] extend TED to a convex formulation by introducing a structured sparsity-inducing norm, and take advantage of a robust sparse representation loss function to suppress outliers. [Shi and Shen, 2016] extend convex TED to a diversity version for selecting complementary samples. More recently, ALFS [Li *et al.*, 2019] study the coupling effect on unsupervised active learning and feature selection, and perform them jointly via the CUR matrix decomposition [Mahoney and Drineas, 2009]. The above methods are linear models, which cannot model nonlinear structures of data in many read-world scenarios. Thus, [Cai and He, 2011] propose a kernel-based method to perform nonlinear sample selection in a manifold adaptive kernel space. However, how to choose appropriate kernel functions for kernel-based methods is usually unclear in practice.

Unlike these approaches, our method explicitly learns a nonlinear embedding by a deep neural network architecture, so that the nonlinear structures of data can be discovered in the latent space, and thus a better representative sample subset can be obtained.

2.2 Matrix Column Subset Selection

Unsupervised active learning is related to one popular mathematical problem: matrix column subset selection (MCSS) [Chan, 1987]. The MCSS problem aims to select a subset of columns from an input matrix, such that the selected columns can capture as much of the input as possible. More precisely, it attempts to select m columns of $\mathbf{X}$ to form a new matrix $\mathbf{Z} \in \mathbf{R}^{d\times m}$ that minimizes the following residual:

$$\min_{\mathbf{Z},\mathbf{C}} ||\mathbf{X} - \mathbf{Z}\mathbf{C}||_\varepsilon = ||\mathbf{X} - \mathbf{Z}\mathbf{Z}^\dagger\mathbf{X}||_\varepsilon$$

where $\mathbf{Z}^\dagger$ is the Moore-Penrose pseudoinverse of $\mathbf{Z}$. $\mathbf{Z}\mathbf{Z}^\dagger$ denotes the projection onto the m-dimensional space spanned by the columns of $\mathbf{Z}$. $\varepsilon = 2$ or F denotes the spectral norm or Frobenius norm.

Different from our method, MCSS still attempts to reconstruct the input based on a linear combination of the selected columns, which cannot model the nonlinearity of data.

3 Deep Unsupervised Active Learning

In this section, we will elaborate the details of the proposed DUAL model for unsupervised active learning. As shown in Figure 1, DUAL mainly consists of the following three blocks: an encoder block and a decoder block are used to learn a nonlinear representation, and a selection block at the junction

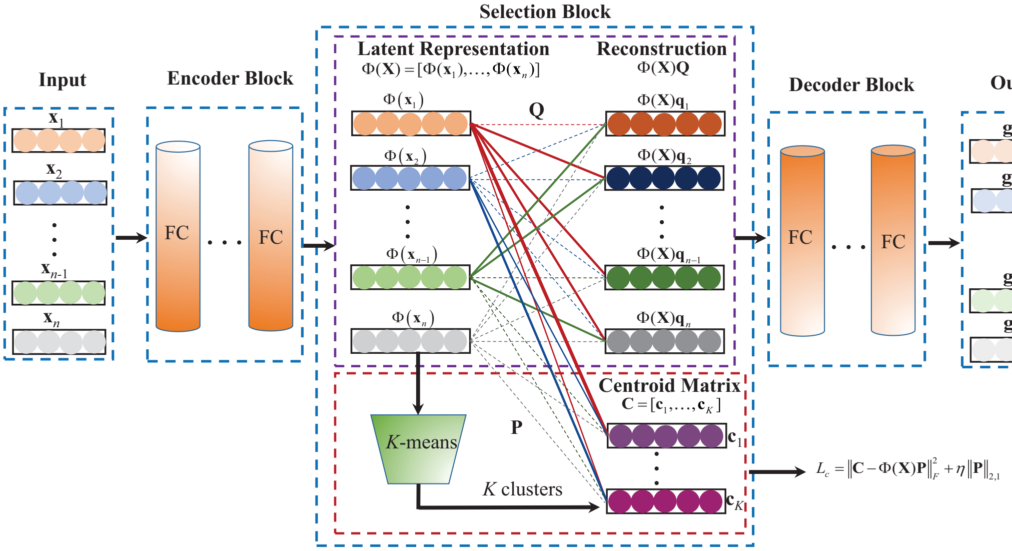


Figure 1: Illustration of the overall architecture. DUAL consists of an encoder block, a selection block and a decoder block. In particular, the encoder and decoder blocks are used to learn a nonlinear representation. The selection block consists of two branches, where each is made up of one fully connected layer without bias and nonlinear activation functions. The top branch attempts to capture as much of the latent representation $\Phi(\mathbf{X})$ as possible, while the bottom one aims to approximate the K cluster centroids well. The dash line in the fully connected layer of the selection block denotes that the sample in the left side has no contribution to the reconstruction of the corresponding sample in the right side, i.e., its weight in $\mathbf{Q}$ or $\mathbf{P}$ is equal to zero.

between the encoder and the decoder used for selecting samples. Before explaining how DUAL is specifically designed for these parts, we first give our problem setting.

3.1 Problem Setting

Let $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_n] \in \mathbb{R}^{d\times n}$ denote a data matrix, where d and n are the feature dimension and the number of training data, respectively. Our goal is to learn a nonlinear transformation Φ to map input $\mathbf{X}$ to a new latent representation $\Phi(\mathbf{X})$, and then select m samples expected to not only approximate the latent representation $\Phi(\mathbf{X})$ well but also preserve the cluster structure of the training data. This problem is quite challenging, since solving it exactly is a hard combinatorial optimization problem (NP-hard). After obtaining a sample subset based on our method DUAL, a prediction model (e.g., a SVM classifier for classification) will be trained by labeling the selected m samples and using $\Phi(\cdot)$ as their new feature representations.

3.2 Encoder Block

In order to learn the nonlinear mapping Φ, we utilize a deep neural network to map each input to a latent space. Since we have no access to the labeled data, we adopt an encoder-decoder architecture because of its effectiveness for unsupervised learning. Our encoder block consists of $L + 1$ layers for performing L nonlinear transformations as the desired Φ (The detail of the decoder block is introduced in Section 3.4). For easy of presentation, we first provide the definition of the output of each layer in the encoder block:

$$\mathbf{h}_i^{(l)} = \sigma(\mathbf{W}_e^{(l)}\mathbf{h}_i^{(l-1)} + \mathbf{b}_e^{(l)}), l = 1, \dots, L, \tag{2}$$

where $\mathbf{h}_i^{(0)} = \mathbf{x}_i, i = 1, \cdots, n$, denotes the original training data $\mathbf{X}$ as the input of the encoder block. $\mathbf{W}_e^{(l)}$ and $\mathbf{b}_e^{(l)}$ are the weights and bias associated with the l-th hidden layer, respectively. $\sigma(\cdot)$ is a nonlinear activation function. Then, we can define our latent representation $\Phi(\mathbf{X})$ as:

$$\Phi(\mathbf{X}) = \mathbf{H}^{(L)} = [\mathbf{h}_1^{(L)}, \cdots, \mathbf{h}_n^{(L)}] \in \mathbb{R}^{d'\times n} \tag{3}$$

where d' denotes the dimension of our latent representation.

3.3 Selection Block

As discussed above, we aim to seek a sample subset which can better capture the whole input patterns and simultaneously preserve the cluster structure of data. To this end, we introduce a selection block at the junction between the encoder and the decoder, as shown in Figure 1. In Figure 1, the selection block consists of two branches, of which each is composed of one fully connected layer but without bias and nonlinear activation functions. The top branch is used to select a sample subset to approximate all samples in the latent space. The bottom one aims to reconstruct the cluster centroids, such that the cluster structure can be well preserved. Next, we will introduce the two branches in detail.

Top branch: In order to best approximate all samples, we present to minimize the following loss function:

$$\mathcal{L}_a = \sum_{i=1}^{n}(||\Phi(\mathbf{x}_i) - \Phi(\mathbf{Y})\mathbf{q}_i||_\ell^2 + \gamma||\mathbf{q}_i||_1) \tag{4}$$
$$s.t.\ \ \Phi(\mathbf{Y}) = [\Phi(\mathbf{y}_1), \dots, \Phi(\mathbf{y}_m)] \in \mathbb{R}^{d'\times m}$$

where $\Phi(\mathbf{Y}) \subset \Phi(\mathbf{X})$ denotes the selected m samples, and $\mathbf{q}_i \in \mathbb{R}^m$ is the reconstruction coefficients for sample $\Phi(\mathbf{x}_i)$.

$||\cdot||_1$ denotes the l_1-norm of a vector. $||\cdot||_\ell$ denotes the ℓ-norm of a vector, indicating certain loss measuring strategy. In this paper, we use a common norm, the l_2-norm, for simplicity. $\gamma \geq 0$ is a tradeoff parameter.

The first term in Eq. (4) aims to pick out m samples to reconstruct the whole dataset in the latent space, while the second term is a regularization term to enforce the coefficient sparse. Unfortunately, there is not an easy solution to (4), as it is a combinatorial optimization. Inspired by [Li *et al.*, 2019], we relax (4) to a convex optimization problem, and write it into a matrix format as

$$\min_{\mathbf{Q}\in\mathbb{R}^{n\times n}} \mathcal{L}_a = ||\Phi(\mathbf{X}) - \Phi(\mathbf{X})\mathbf{Q}||_F^2 + \gamma||\mathbf{Q}||_{2,1} \quad s.t.\ ||diag(\mathbf{Q})||_1 = 0 \tag{5}$$

where $||\cdot||_{2,1}$ denotes the $l_{2,1}$-norm of a matrix, defined as sum of the l_2-norms of row vectors. The constraint condition ensures the diagonal elements of $\mathbf{Q}$ equal to zeros, avoiding the degenerated solution. To minimize (5), we utilize the fact that, $\mathbf{Q} = [\mathbf{q}_1, \ldots, \mathbf{q}_n]$ can be thought of the parameters of a fully connected layer without bias and nonlinear activations, such that Φ and $\mathbf{Q}$ can be solved jointly through a standard backpropagation procedure.

Bottom branch: In the top branch, the selected samples can approximate the whole dataset well, while it might fail to preserve the cluster structure of data. To solve this problem, we first utilize a clustering algorithm to cluster the data into some clusters in the latent space, and then select a sample subset to best approximate the obtained cluster centroids.

Specifically, given the latent representation $\Phi(\mathbf{X})$, we can obtain K clusters based on a clustering algorithm, K-means used in this paper. Then, we denote the cluster centroid matrix $\mathbf{C}$ as

$$\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_K] \in \mathbb{R}^{d'\times K} \tag{6}$$

where $\mathbf{c}_k$ denotes the k-th cluster centroid, and K is the number of clusters.

In order to preserve the cluster structure, we minimize the following loss function:

$$\mathcal{L}_c = ||\mathbf{C} - \Phi(\mathbf{X})\mathbf{P}||_F^2 + \eta||\mathbf{P}||_{2,1} \tag{7}$$

where $\mathbf{P}$ is the coefficient matrix for reconstructing $\mathbf{C}$. $\eta \geq 0$ is a tradeoff parameter. Similarly, $\mathbf{P}$ can also be thought of the parameters of a fully connected layer without bias and nonlinear activation functions, and thus can be optimized jointly with Φ.

3.4 Decoder Block

As aforementioned, since we have no access to the label information, we attempt to recover each input by a decoder block, to guide the learning of the latent representation. In other words, each input actually plays the role of a supervisor. Similar to the encoder block, our decoder block also consists of $L+1$ layers for performing L nonlinear transformations. The output of each layer in the decoder block can be defined as:

$$\mathbf{g}_i^{(l)} = \sigma(\mathbf{W}_d^{(l)}\mathbf{g}_i^{(l-1)} + \mathbf{b}_d^{(l)}), l = 1, \ldots, L, \tag{8}$$

where $\mathbf{g}_i^{(0)} = \Phi(\mathbf{X})\mathbf{q}_i$ means that the output of the selection block is used as the input of the decoder block. $\mathbf{W}_d^{(l)}$ and $\mathbf{b}_d^{(l)}$ are the weights and bias associated with the l-th hidden layer in the decoder block, respectively.

Then, the reconstruction loss function is defined as:

$$\mathcal{L}_r = \sum_{i=1}^{n} ||\mathbf{x}_i - \mathbf{g}_i^{(L)}||_2^2 = ||\mathbf{X} - \mathbf{G}^{(L)}||_F^2, \tag{9}$$

where $\mathbf{g}_i^{(L)}$ denotes the output of the decoder to the input $\mathbf{x}_i$, and $\mathbf{G}^{(L)}$ is expressed by $\mathbf{G}^{(L)} = [\mathbf{g}_1^{(L)}, \mathbf{g}_2^{(L)}, \ldots, \mathbf{g}_n^{(L)}]$.

3.5 Overall Model and Training

After introducing all the building blocks of this work, we now give the final training objective and explain how to jointly optimize it. Based on the Eq. (5), (7), and (9), the final loss function is defined as:

$$\min\ \mathcal{L} = \mathcal{L}_r + \alpha\mathcal{L}_a + \beta\mathcal{L}_c \tag{10}$$

where α and β are two positive tradeoff parameters.

In (10), there are three reconstruction loss terms. The first term denotes the reconstruction loss of the encoder-decoder model in Eq. (9). The second term corresponds to input patterns reconstruction loss in Eq. (5). The last term is the cluster centroids reconstruction loss as shown in Eq. (7).

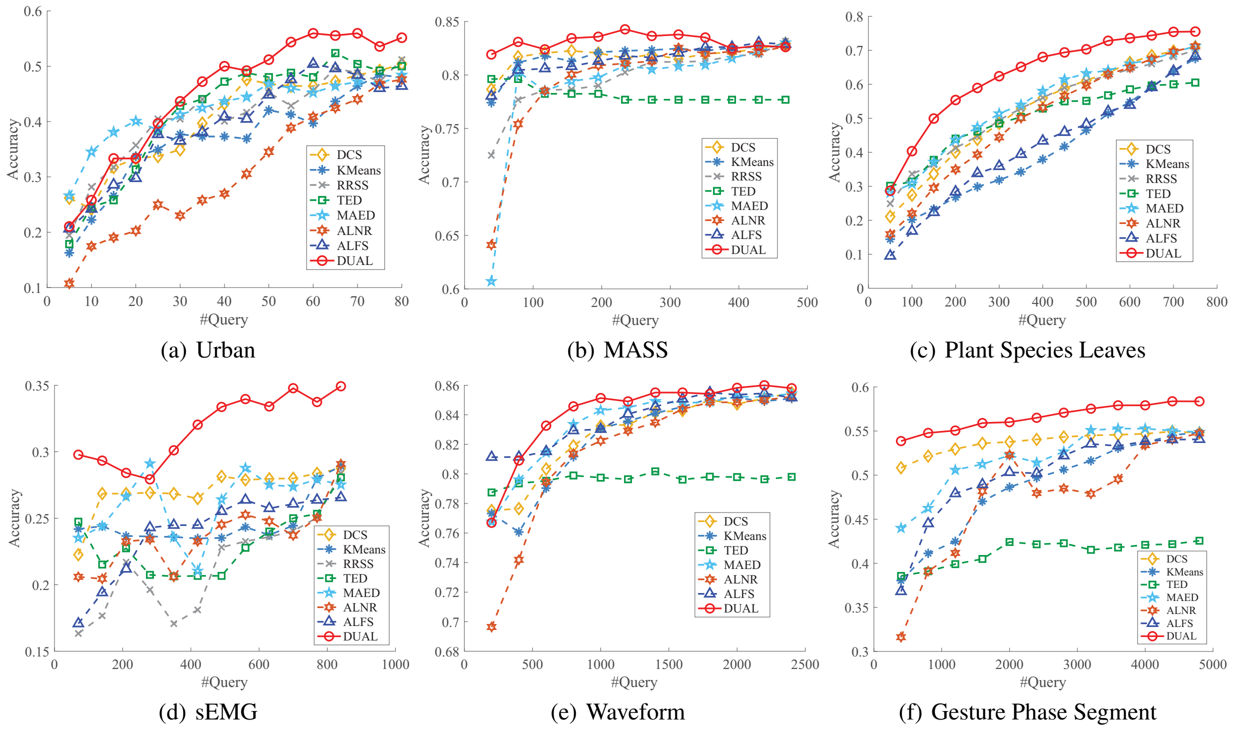
To solve (10), we present a three-stage training strategy which is an end-to-end trainable fashion. Firstly, we pre-train the encoder and decoder block in the beginning without considering the selection block. After that, we utilize the output of the encoder block as the latent representation to perform K-means, and regard the obtained K cluster centroids as the centroid matrix $\mathbf{C}$ for subsequent sample selection. Lastly, we use the pre-trained parameters to initialize the encoder and decoder blocks, and load all data into a batch to optimize the whole network, i.e., minimizing the loss (10). Throughout the experiment, we use three fully connected layers in the encoder and decoder blocks, respectively. The rectified linear unit (ReLU) is used as the non-linear activation function. In addition, we use Adam [Kingma and Ba, 2014] as the optimizer, where the learning rate is set to 1.0×10^{-4}.

Once the model is trained, we can obtain two reconstruction coefficient matrices $\mathbf{Q}$ and $\mathbf{P}$. We then use a simple strategy to select the most representative samples based on $\mathbf{Q}$ and $\mathbf{P}$. Specifically, we calculate the l_2-norm of the rows of $\mathbf{Q}$ and $\mathbf{P}$ respectively, and obtain two corresponding vectors $\hat{\mathbf{q}}$ and $\hat{\mathbf{p}}$. After that, we normalize $\hat{\mathbf{q}}$ and $\hat{\mathbf{p}}$ to the range of [0,1]. Finally, we can sort all the samples by adding normalized $\hat{\mathbf{q}}$ to normalized $\hat{\mathbf{p}}$ in descending order, and select the top m samples as the most representative ones.

4 Experiments

To verify the effectiveness of our method DUAL, we perform the experiments on six publicly available datasets which are widely used for active learning [Baram *et al.*, 2004]. The details of these datasets are summarized in Table 1 [1].

[1] These datasets are downloaded from the UCI Machine Learning Repository



(a) Urban

(b) MASS

(c) Plant Species Leaves

(d) sEMG

(e) Waveform

(f) Gesture Phase Segment

Figure 2: Comparisons of different active learning methods in terms of accuracy on six benchmark datasets.

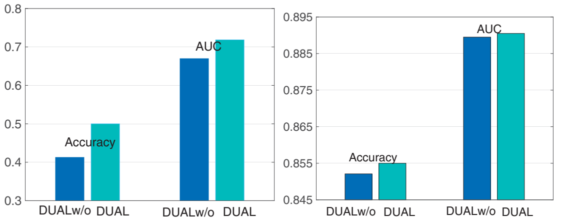
Dataset	Size	Dimension	Class
Urban Land Cover	168	148	9
Waveform	5000	40	3
sEMG	1800	3000	6
Plant Species Leaves	1600	64	100
Gesture Phase Segment	9873	50	5
Mammographic Mass	961	6	2

Table 1: Summary of Experimental Datasets.

4.1 Experimental Setting

Compared methods[2]: We compare DUAL with several typical unsupervised active learning algorithms, including TED [Yu *et al.*, 2006], RRSS [Nie *et al.*, 2013], ALNR [Hu *et al.*, 2013], MAED [Cai and He, 2011], ALFS [Li *et al.*, 2019]. We also compare Deterministic Column Sampling(DCS) [Papailiopoulos *et al.*, 2014] in our experiments. In addition, we also take K-means as a baseline.

Experimental protocol: Following [Li *et al.*, 2019], for each dataset, we randomly select 50% of the samples as candidates for sample selection, and use the rest as the testing data. To evaluate the effectiveness of sample selection, we train a SVM classifier with a linear kernel and $C = 100$ by using these selected samples as the training data. We set the parameters $\gamma = \eta$ for simplicity, and search all tradeoff parameters in our algorithm from $\{0.01, 0.1, 1, 10\}$. The number of clusters K is searched from $\{5, 10, 20, 50\}$. We use accuracy and AUC to measure the performance. We repeat every test case five times, and report the average result.



(a) Urban

(b) Waveform

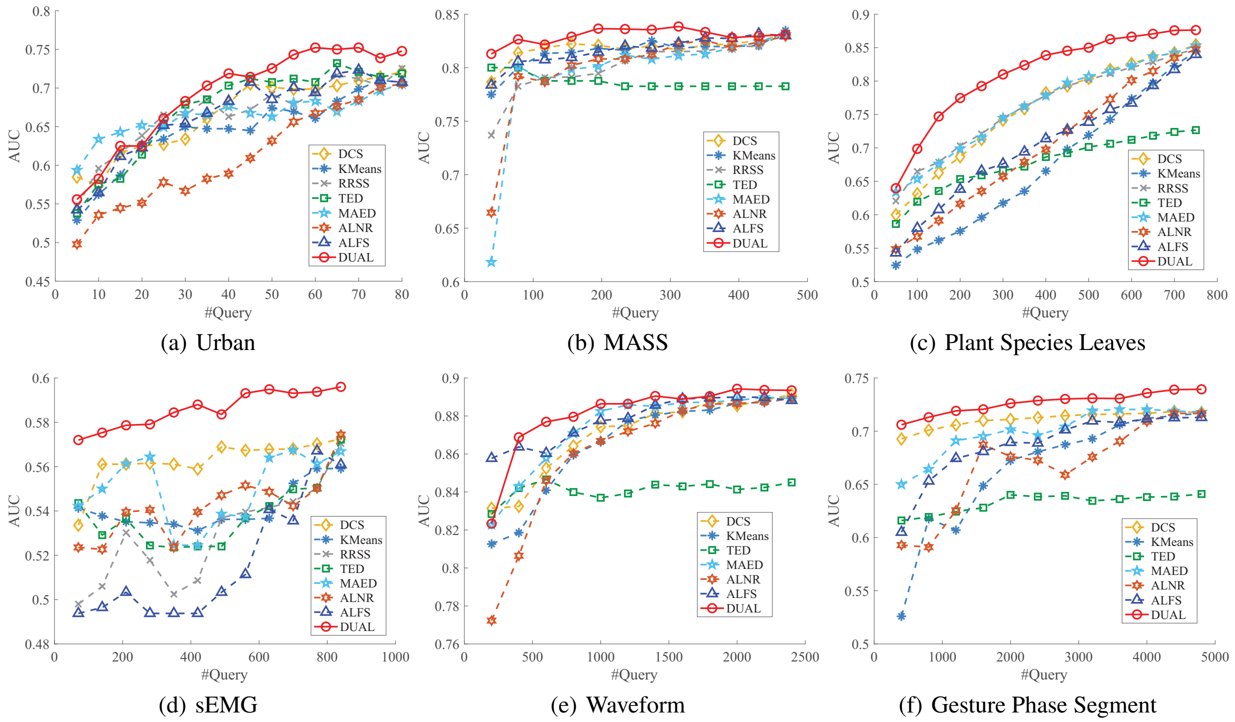
Figure 4: The effectiveness verification of the components in the selection block.

4.2 Experimental Result

General Performance: Figure 2 and 3 show the scores in terms of different numbers of queries. Our method outperforms other algorithms under most of cases, especially on the larger datasets. This illustrates that by preserving input patterns and cluster structures, DUAL can select representative samples. In addition, we observe that MAED and DCS have good results on some datasets. This may be because MAED is a nonlinear method that can handle complex data, while D-CS selects a subset with largest leverage scores, resulting in a good low-rank matrix surrogate.

Ablation Study: We study the effectiveness of the components in our selection block on the Urban and Waveform datasets. The experimental setting is as follows: we only con-

[2] All source codes are obtained from the authors of the corresponding papers, except K-means and ALNR.



(a) Urban (b) MASS (c) Plant Species Leaves

(d) sEMG (e) Waveform (f) Gesture Phase Segment

Figure 3: Comparisons of different active learning methods in terms of AUC on six benchmark datasets.

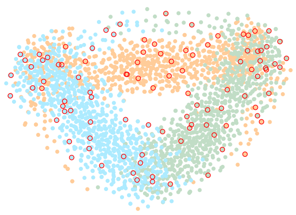


Figure 5: The visualization by t-SNE. The red circles denote the selected samples, and other color solid circles denote different classes.

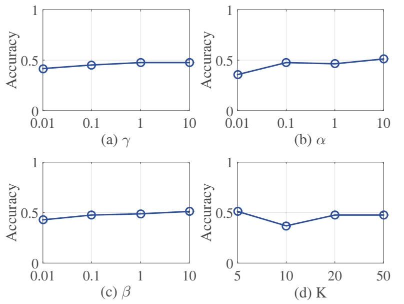


Figure 6: Parameter Study on the Urban dataset.

sider to preserve the whole input patterns, i.e., setting $\beta = 0$ in (10). We call it DUALw/o. The number of queries is set to half of the candidates. Figure 4 shows the results. DUAL achieves better results than DUALw/o, which indicates that preserving cluster structures is helpful for unsupervised active learning. On the Waveform dataset, the improvement of DUAL over DUALw/o is a little bit light. This is because the numbers of samples among different classes are balanced in this dataset, and thus selecting samples by only modeling input patterns may preserve the cluster structure well.

Visualization: In this subsection, we apply our method DUAL on the Waveform dataset to give an intuitive result. We use the t-SNE [Maaten and Hinton, 2008] to visualize the samples selected by our method, as shown in Figure 5. The samples selected by DUAL can better represent the dataset. This is because DUAL can better capture the nonlinear structure of data by performing active learning in the latent space.

Parameter Study: We study the sensitivity of our algorithm about the tradeoff parameters α, β, $\gamma(=\eta)$, and the number of clusters K on the Urban dataset. We fix the number of queries to half of the candidates, and report their accuracies. The results are shown in Figure 6. Our method is not sensitive to the parameters with a relatively wide range.

5 Conclusion

In this paper, we proposed a deep learning based framework for unsupervised active learning, called DUAL. DUAL can model the nonlinear structure of data, and select representative samples to preserve the input pattern and the cluster structure of data. Extensive experimental results on six datasets demonstrate the effectiveness of our method.

Acknowledgements

This work was supported by the NSFC Grant No. 61806044, 61932004, N181605012, and 61732003.

References

[Aggarwal *et al.*, 2014] Charu C Aggarwal, Xiangnan Kong, Quanquan Gu, Jiawei Han, and S Yu Philip. Active learning: A survey. In *Data Classification*, pages 599–634. Chapman and Hall/CRC, 2014.

[Baram *et al.*, 2004] Yoram Baram, Ran El Yaniv, and Kobi Luz. Online choice of active learning algorithms. *JMLR*, 5(3):255–291, 2004.

[Cai and He, 2011] Deng Cai and Xiaofei He. Manifold adaptive experimental design for text categorization. *TKDE*, 24(4):707–719, 2011.

[Cai *et al.*, 2019] Jia-Jia Cai, Jun Tang, Qing-Guo Chen, Yao Hu, Xiaobo Wang, and Sheng-Jun Huang. Multi-view active learning for video recommendation. In *IJCAI*, pages 2053–2059. AAAI Press, 2019.

[Chan, 1987] Tony F Chan. Rank revealing qr factorizations. *Linear algebra and its applications*, 88:67–82, 1987.

[Elhamifar *et al.*, 2013] Ehsan Elhamifar, Guillermo Sapiro, Allen Yang, and S. Shankar Sasrty. A convex optimization framework for active learning. In *ICCV*, pages 209–216, 2013.

[Freund *et al.*, 1997] Yoav Freund, H. Sebastian Seung, Eli Shamir, and Naftali Tishby. Selective sampling using the query by committee algorithm. *Machine Learning*, 28(2-3):133–168, 1997.

[FY *et al.*, 2015] Zhu FY, Xingliang Zhu, Xiang SM, Pan CH, et al. 10,000+ times accelerated robust subset selection (arss). In *AAAI*, pages 3217–3223, 2015.

[Haussmann *et al.*, 2019] Manuel Haussmann, Fred Hamprecht, and Melih Kandemir. Deep active learning with adaptive acquisition. In *IJCAI*, pages 2470–2476, 2019.

[Hu *et al.*, 2013] Yao Hu, Debing Zhang, Zhongming Jin, Deng Cai, and Xiaofei He. Active learning via neighborhood reconstruction. In *IJCAI*, pages 1415–1421, 2013.

[Huang *et al.*, 2010] ShengJun Huang, Rong Jin, and Zhi-Hua Zhou. Active learning by querying informative and representative examples. In *NIPS*, pages 892–900, 2010.

[Jain and Kapoor, 2009] Prateek Jain and Ashish Kapoor. Active learning for large multi-class problems. In *CVPR*, pages 762–769, 2009.

[Joshi *et al.*, 2009] Ajay J Joshi, Fatih Porikli, and Nikolaos Papanikolopoulos. Multi-class active learning for image classification. In *CVPR*, pages 2372–2379, 2009.

[Kapoor *et al.*, 2007] Ashish Kapoor, Eric Horvitz, and Sumit Basu. Selective supervision: Guiding supervised learning with decision-theoretic active learning. In *IJCAI*, volume 7, pages 877–882, 2007.

[Kingma and Ba, 2014] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[Li *et al.*, 2017] Qin Li, Xiaoshuang Shi, Linfei Zhou, Zhifeng Bao, and Zhenhua Guo. Active learning via local structure reconstruction. *PRL*, 92:81–88, 2017.

[Li *et al.*, 2019] Changsheng Li, Xiangfeng Wang, Weishan Dong, Junchi Yan, Qingshan Liu, and Hongyuan Zha. Joint active learning with feature selection via cur matrix decomposition. *TPAMI*, 41(6):1382–1396, 2019.

[Maaten and Hinton, 2008] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using tsne. *JMLR*, 9(11):2579–2605, 2008.

[Mahoney and Drineas, 2009] Michael W Mahoney and Petros Drineas. Cur matrix decompositions for improved data analysis. *PNAS*, 106(3):697–702, 2009.

[Nie *et al.*, 2013] Feiping Nie, Wang Hua, Heng Huang, and Chris Ding. Early active learning via robust representation and structured sparsity. In *IJCAI*, pages 1572–1578, 2013.

[Papailiopoulos *et al.*, 2014] Dimitris Papailiopoulos, Anastasios Kyrillidis, and Christos Boutsidis. Provable deterministic leverage score sampling. In *KDD*, pages 997–1006, 2014.

[Roweis and Saul, 2000] Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500):2323–2326, 2000.

[Settles, 2009] Burr Settles. Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2009.

[Shi and Shen, 2016] Lei Shi and Yi-Dong Shen. Diversifying convex transductive experimental design for active learning. In *IJCAI*, pages 1997–2003. AAAI Press, 2016.

[Vijayanarasimhan and Grauman, 2014] Sudheendra Vijayanarasimhan and Kristen Grauman. Large-scale live active learning: Training object detectors with crawled data and crowds. *IJCV*, 108(1-2):97–114, 2014.

[Yu *et al.*, 2006] Kai Yu, Jinbo Bi, and Volker Tresp. Active learning via transductive experimental design. In *ICML*, pages 1081–1088, 2006.

[Zhang *et al.*, 2011] Lijun Zhang, Chun Chen, Jiajun Bu, Deng Cai, Xiaofei He, and Thomas S Huang. Active learning based on locally linear reconstruction. *TPAMI*, 33(10):2026–2038, 2011.

[Zhang *et al.*, 2017] XiaoYu Zhang, Shupeng Wang, and X-iaochun Yun. Bidirectional active learning: A two-way exploration into unlabeled and labeled data set. *TNNLS*, 26(12):3034–3044, 2017.

[Zheng and Ye, 2015] Wang Zheng and Jieping Ye. Querying discriminative and representative samples for batch mode active learning. *ACM Transactions on Knowledge Discovery from Data*, 9(3):1–23, 2015.