Uncertainty and Utility Sampling with Pre-Clustering

Zhixin Huang, Yujiang He, Stephan Vogt, and Bernhard Sick

Intelligent Embedded Systems, University of Kassel, Kassel, Germany
{zhixin.huang,yujiang.he,stephan.vogt,bsick}@uni-kassel.de

Abstract. Uncertainty sampling is one of the main approaches in deep active learning. In the early phase of uncertainty sampling, uninformative instances are usually selected due to missing exploration of the data space. This can result in a poor quality model leading to poorer acquisitions and further leading to a poorer model. Clustering algorithms can analyze large amounts of unlabeled data in an unsupervised way. A cluster center can be seen as the representative of its cluster and is often highly useful for querying the label from the oracle. Therefore, we propose an algorithm that enables the model to explore the data space at the initial stage using pre-clustering, and enhances the exploration of uncertainty sampling continually based on a combination of uncertainty and utility metrics. The preliminary experimental results show that the proposed algorithm supports balance and imbalanced data scenarios. Besides, our algorithm can achieve a higher classification accuracy compared to baselines methods, even under fewer annotations.

Keywords: Active Learning · Deep Active Learning · Bayesian Neural Network · Uncertainty Sampling · Clustering.

1 Introduction

Deep learning (DL) has a strong learning ability to process high-dimensional data and extract features automatically [24], while DL is often very greedy for data [11]. Active learning is concerned with reducing annotation costs effectively and ensuring a predetermined level of accuracy. However, a major challenge in AL is its lack of scalability to high-dimensional data [29]. Therefore, an approach that combines DL and AL will significantly expand their application potential. This combined approach, referred to as deep active learning (DAL), mainly contains two parts: the AL query strategy on the unlabeled data set and the DL model training [24]. In the pool-based AL scenario, the selection strategy chooses the best sample based on the evaluation and ranking of the entire large data set. The annotated samples are used to train the model and improve the data acquisition for the next AL iteration. The uncertainty-based approach is one of the most common pool-based methods in the application, because it is simple in form and has low computational complexity [24]. Many DAL [1, 10, 22, 23] methods use the uncertainty sampling (US) strategy directly. However, there are still two challenges that have to be overcome:

- **Unreliable uncertainty at the initial AL phase** Uninformative instances are usually selected based on unreliable uncertainty due to an unclear sense of the data space at the early stage. This can result in a poor quality model leading to poorer acquisition and further leading to a poorer model [3].
- **Uncertainty sampling lacks exploration** For uncertainty sampling in DAL context, [6, 14, 12] utilize batch acquisition and query the top n instances with the highest scores. However, it is likely to select a set of information-rich but similar samples [33]. It leads to insufficient exploration, i.e., the knowledge regarding the data distribution is not fully utilized [24], which makes low DL model training efficiency and high annotation cost.

To address the first challenge, it is crucial to find the most representative instances from the large unlabeled data set at the initial AL phase. The general method [7] is to use random selection (RS) at the beginning of the training process for exploration. However, this method could fail for imbalanced data set because the selected instances are less representative, and most of them locate dense areas [30]. The model can deeply learn the true data space only when sufficient labels of data are available. However, it will increase annotation cost. Unsupervised learning algorithms can analyze large amounts of unlabeled data. For example, the K-Means [26] algorithm is one of the most common clustering algorithms for knowledge discovery in data mining. The cluster information is helpful for AL in two aspects: (1) The instances located in the center of clusters are more representative than the others and should be labeled firstly; (2) Samples in the same cluster are likely to have the same label [21].

For the second challenge, a feasible solution is to use a hybrid query strategy to enhance the exploration of US. The similarity between samples is a method [21, 15] to measure the similarity amongst instances by calculating the feature vectors' distance between each other. Similar to US, these algorithms are often only good at exploitation, i.e., the learners tend to only focus on instances near the current decision boundary [24]. But in the opposite direction, we can also utilize the similarity to exclude similar samples. After sorting a batch of instances based on the uncertainty through US, we could filter out similar instances to improve the exploration of selection strategy.

To overcome the challenges mentioned above, the two core ideas of our proposed algorithm are: (1) At the initial phase, we label the instances closest to cluster centers to train the model for estimating reliable uncertainty. (2) The selection of the most informative instance depends on two strategies, uncertainty and utility. The uncertainty evaluates the epistemic uncertainty of Bayesian Neural Network (BNN) [6, 7] to an instance. The utility filters out the instances which are similar to the already labeled instances. Since US lacks exploration in the data space, the utility metric helps the model discover some valuable instances far away from the current decision boundary. Therefore, we propose our algorithm **U**ncertainty and **U**tility sampling with **P**re-**C**lustering (UUPC). Compared to the baselines, our algorithm can achieve a higher classification accuracy under fewer annotations.

The remainder of this article starts with a summary of the related work. The details of the algorithm and the experiments are introduced in Section 3 and 4 respectively. This article is closed with a conclusion and an outlook on our future work in this field.

2 Related Work

The uncertainty-based query strategies (e.g., Margin Sampling and Entropy) in the DAL scenario are widely used [6, 14, 12] because it is convenient to combine with the output of the DL model. Traditional DL requires a large amount of labeled data to obtain reliable uncertainty estimation. In the DAL scenario with large unlabeled data, epistemic uncertainty is particularly valuable because it allows the model to assess its lack of knowledge. For this reason, a method that combines deep Bayesian neural network with US has been proposed [7, 12, 22]. However, as analyzed in Section 1, US could select uninformative instances at the initial AL phase and lack exploration. Therefore, some hybrid query strategies are developed [32, 34], taking into account the uncertainty and diversity of samples. Exploration-P [32] utilizes a deep neural network to obtain the uncertainty and the similarity between the samples. Besides, this method uses RS strategy for exploration purposes in the early AL phase. The combination of AL and K-means clustering has been researched in previous works [13, 21] to find the most representative instances. DBAL [34] presents a hybrid query approach that utilizes the K-means clustering algorithm to explore the diversity of instances in each mini-batch. Contrary to [34], which performs clustering in each AL iteration, our approach annotates labels based on cluster centers only at the initial AL phase to pretrain the BNN model. Thus, it can avoid labeling samples repeatedly in the same cluster. Similar to select the most representative instances by clustering, the core set approach is also a representative query strategy. The basic idea is constructing a core set to represent the distribution of the feature space of the entire original data set, thereby reducing the labeling cost of AL [27, 31]. However, the core-set approach requires building a large distance matrix on the unlabeled data set, the search process is computationally expensive especial on the large data set [2].

3 Problem Formulation and Algorithms

In the general classification, one sample is described by $\mathbf{x} \in \mathcal{X}$ and its label from C classes with a corresponding label $y \in \mathcal{Y} = \{1, \cdots, C\}$. The clustering information can be described explicitly by introducing the cluster label $k \in \{1, \cdots, K\}$, where K is the number of clusters in the data. In the pool-based AL, we define $\mathcal{U} = \{\mathbf{x}_1, \cdots, \mathbf{x}_N\}$ as an unlabeled set with N samples. Labels are not available at the beginning but can be annotated by the oracle. The query strategy selects an instance $\mathbf{x} \in \mathcal{U}$ and asks the oracle for the corresponding label $y \in \mathcal{Y}$. The newly labeled instance is removed from the unlabeled set $\mathcal{U} \leftarrow \mathcal{U} \backslash \mathbf{x}$. We add the instance with its label to the labeled set $\mathcal{L} \leftarrow \mathcal{L} \cup (\mathbf{x}, y)$, and train supervised learning models such as SVM and DNN on $\mathcal{L}$.

3.1 Pre-Clustering at initial AL Phase

Selecting the most representative instances from the unlabeled data by labeling cluster centers is heavily dependent on the quality of clustering results. In K-Means, the crucial parameter that affects the goodness of clustering results is the number of clusters, which should be optimized. The evaluation without any labels must be performed using the model itself. The elbow method [16] is the most popular heuristic approach, which calculates the sum of squared distances (SD) from each point to its assigned center. The unsupervised evaluation scores such as Silhouette Coefficient (SC) [25], Calinski-Harabasz Index (CHI) [4] and Davies-Bouldin Index (DBI) [9] could also be applied to the elbow method. We will calculate multiple cluster scores to determine the optimal number of clusters K_o. To optimize SC and CHI, we have to maximize the scores, while lower SD and DBI indicate a model with better defined clusters so they must be minimized. We take the reciprocal of SC and CHI to unify the optimization direction. The weighted score of pre-clustering (PC) is calculated by following:

$$\begin{aligned}\underset{\text{PC}}{\text{Score}}\,(K,\mathcal{U}) = \;& \alpha_1\, \text{SD}\,(K,\mathcal{U}) + \alpha_2\, \text{DBI}\,(K,\mathcal{U}) \\ & + \alpha_3 \frac{1}{\text{SC}\,(K,\mathcal{U})} + \alpha_4 \frac{1}{\text{CHI}\,(K,\mathcal{U})} + \lambda K.\end{aligned} \tag{1}$$

The weights of each score are $\alpha_{1,\cdots,4}$, and the sum is 1. The $\alpha_{1,\cdots,4}$ could be selected by expert knowledge, or in the absence of detailed expert knowledge, like in the experiment in Section 4, all weights are selected to be the same value. In our definition, K must be equal or greater than C. For example, MNIST [19] requires at least 10 clusters, one per class. K_{max} indicates the maximum budget of annotations at the initial AL phase, and we expect that $C < K_{max} \ll N$. Since the above four cluster evaluation scores have different scales, in practice, we calculate a set for each type of score (SD, DBI and reciprocal of SC and CHI) from C to K_{max} and normalize each set to 0-1 range. Then we add the four scores to obtain a set of $\underset{\text{PC}}{\text{Score}}\,(K,\mathcal{U})$, where $K \in \{C, \ldots, K_{max}\}$. The larger the K, the smaller the $\underset{\text{PC}}{\text{Score}}$, which means that the more refined clustering. However, the labeling cost must be considered because the oracle has to annotate every instance closest to the center in each cluster. Therefore we append λK into $\underset{\text{PC}}{\text{Score}}\,(K,\mathcal{U})$ as the regularization, where λ is the weight of regularization and proportional to the cost of an annotation. Setting a proper value of λ is dependent on the application scenario and requires expert experience. The Bayesian information criterion (BIC) and the Akaike information criterion (AIC) could determine the appropriate number of clusters without tuning regularization [28, 8]. But they can be applied only if we extend the clustering algorithm beyond K-Means to Gaussian Mixture Model (GMM). Since this paper utilizes pre-clustering by K-Means, BIC and AIC will be researched in future work. The optimal number of clusters K_o can be described as follows:

$$K_o = \underset{K \in \{C, \cdots, K_{max}\}}{\operatorname{argmin}} \underset{\mathrm{PC}}{\operatorname{Score}} (K, \mathcal{U}) \tag{2}$$

Assume that the information about the class label y is encoded in the cluster k. The set of elements in cluster k is $\mathbf{c}_k$. Once the data probability distribution of clusters $p(\mathbf{x} \in \mathbf{c}_k)$ and the class y_k of each cluster center $\mathbf{x}_k$ are known, we can infer the probability distribution of class $p(y|\mathbf{x} \in \mathbf{c}_k)$ with respect to all samples in $\mathbf{c}_k$ [21]. However, using the cluster center to annotate all instances' labels is not reliable because the samples located at the intersection of clusters are easily misclassified. In contrast with refining smaller clusters [21], our method only uses the pre-clustering to pretrain the model. In detail, we only label the instances closest to each cluster center by oracle $p(y_k|\mathbf{x}_k, k)$ and put them into the labeled data set $\mathcal{L} = \{(\mathbf{x}_j, y_j) \mid j \in \{1, \cdots, K_o\}\}$, where K_o is optimal number of clusters. At the initial phase of our approach, the BNN will learn the initial labeled data set to get optimal posterior parameters for reliable uncertainty estimation. Then the oracle will label the most informative instances based on the combination of the following two selection functions: uncertainty and utility.

3.2 Uncertainty-Utility Selection Strategy at AL Phase

The BNN can be defined as $f(\mathbf{x}, \boldsymbol{\theta})$. $p(\boldsymbol{\theta})$ where $\boldsymbol{\theta} \in \Theta$ is a prior on the parameter space Θ. The likelihood $p(y|\mathbf{x}, \boldsymbol{\theta})$ is determined by $\operatorname{softmax}(f(\mathbf{x}, \boldsymbol{\theta}))$. The goal is to obtain the posterior distribution over $\boldsymbol{\theta}$ from labeled training set $\mathcal{L}$:

$$p(\boldsymbol{\theta}|\mathcal{L}) = \frac{p(\mathcal{L}|\boldsymbol{\theta})\, p(\boldsymbol{\theta})}{p(\mathcal{L})} \tag{3}$$

The $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_T$ are sampled T times to get an monte carlo estimate of the predictive probabilitiy distribution on the label y as the average regarding a new unlabeled instance $\mathbf{x}^* \in U$:

$$\hat{p}(y|\mathbf{x}^*, \mathcal{L}) = \frac{1}{T} \sum_{t=1}^{T} p(y|\mathbf{x}^*, \mathcal{L}, \boldsymbol{\theta}_t) \tag{4}$$

Equation 4 describes the general uncertainty estimation of BNN, and it includes both the epistemic and aleatoric uncertainty of the prediction y. In our case, we calculate the entropy over the predicted class probabilities of a new instance to estimate the uncertainty score as given in the numerator of Eq. 5. In each AL iteration, the scores of instances in $\mathcal{U}$ are normalized into a 0–1 range, where 1 is the most uncertain score, indicating that being annotated is often very useful. The function $\operatorname{Uncr}(\mathbf{x}^*)$ can evaluate the uncertainty score for each instance in $\mathcal{U}$:

$$\operatorname{Uncr}(\mathbf{x}^*) = \frac{-\sum_{c}^{C} \hat{p}(y = c|\mathbf{x}^*, \mathcal{L}) \log_2 (\hat{p}(y = c|\mathbf{x}^*, \mathcal{L}))}{\log_2 (C)}. \tag{5}$$

As mentioned in Section 2, the uncertainty metric requires to be enhanced with exploration of the data space. Although in the initial stage, we use pre-clustering to help BNN to obtain reliable uncertainty estimations quickly, some valuable instances are far from the current existing decision boundary. Therefore, we define a utility metric to enhance the exploration of US continually. We define the Euclidean distances between two instances $\mathbf{x}_1$ and $\mathbf{x}_2$ as $\mathrm{Dis}\left(\mathbf{x}_1, \mathbf{x}_2\right)$. The similarity between the instance $\mathbf{x}^*$ to class c is defined as the median distances of $\mathbf{x}^*$ to all instances of c in the $\mathcal{L}$. The formulation can be written as:

$$\mathrm{Sim}\left(\mathbf{x}^*, c\right) = \mathrm{median}(\{\mathrm{Dis}\left(\mathbf{x}^*, \mathbf{x}\right),\ \text{where}\ \left(\mathbf{x}, y\right) \in \mathcal{L} \text{ and } y = c\}). \tag{6}$$

The standard deviation of the similarities between the instance and each class represents the trend of which class it belongs to. The higher standard deviation indicates the instance is likely to be classified to one single class. When the standard deviation is lower, the instance is located in the intersection of multiple classes, and annotation by the oracle could be more beneficial. For a paired comparison with uncertainty, we transfer the optimization task of this score into a maximization problem. The scale of uncertainty score is 0-1. Hence in practice, we calculate the utility score of each instance in a batch and normalize the entire batch of utility scores to the same scale. Eq. 7 shows the method of calculating the utility of a single instance $\mathbf{x}^*$.

$$\mathrm{Utility}(\mathbf{x}^*) = \frac{1}{\mathrm{std}\left(\{\mathrm{Sim}\left(\mathbf{x}^*, c_1\right), \cdots \mathrm{Sim}\left(\mathbf{x}^*, c_C\right)\}\right)} \tag{7}$$

Uncertainty-utility (UU) score is defined as follows:

$$\underset{\mathrm{UU}}{\mathrm{Score}}\left(x*\right) = \gamma_1\, \mathrm{Uncr}\left(\mathbf{x}^*\right) + \gamma_2\, \mathrm{Utility}\left(\mathbf{x}^*\right) \tag{8}$$

where γ_1 and γ_2 are in 0-1 range and control the weights of two selection metrics separately. The weights could be selected by expert knowledge, or in the absence of detailed expert knowledge, γ_1 and γ_2 are each selected equal to 1. The higher score, indicating the more worthy of being annotated.

3.3 Batch-based UUPC Algorithm

With batch training, our method could have more efficient training on large data sets: (1) Clustering, such as K-Means, passes through the entire data set to obtain the centers. The training process is time-consuming, which is proportional to the amount of data. The mini-batch-based K-Means [26] uses a batch-based method to cluster large data sets to reduce computation costs. (2) For traditional uncertainty sampling, each iteration requires uncertainty estimation for all instances in $\mathcal{U}$. In DAL scenario, we use batch-based sample querying to improve training efficiency [24].

At each acquisition step, we score a batch of candidate unlabeled samples $\mathcal{B} \subseteq \mathcal{U}$, where $\mathcal{B} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_b\}$ and b refers to the batch size. Based on the $\underset{\mathrm{UU}}{\mathrm{Score}}$,

Algorithm 1 UUPC Algorithm for Batch Training

Input: Unlabeled data set $\mathcal{U} \leftarrow \mathcal{X}$, initial labeled set $\mathcal{L} \leftarrow \emptyset$, one batch data $\mathcal{B} \subseteq \mathcal{U}$ with b samples is selected randomly, the process of batch sampling is described as $\text{BatchSampling}\left(\mathcal{U}, b\right)$, the maximum number of AL iterations for pre-clustering phase B_{pc} and for UU sampling phase B_{uu}, N_{uu} instances are annotated per batch.

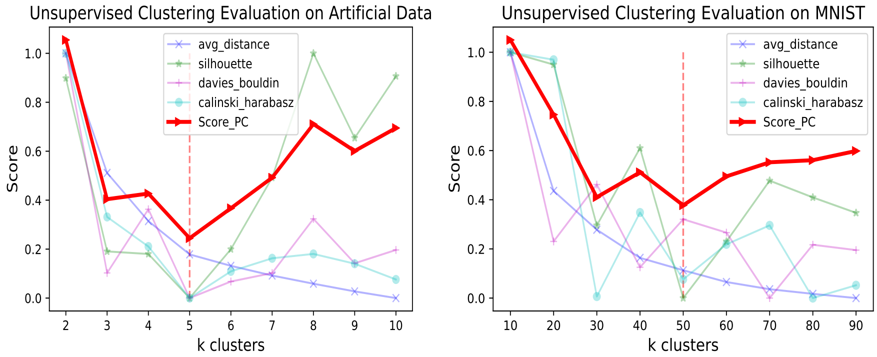
Output: Optimized number of cluster K_o, labeled data set $\mathcal{L}$, BNN model $f\left(\mathbf{x}, \boldsymbol{\theta}\right)$

```
 1: $K_o \leftarrow \underset{K \in \{C, \cdots K_{max}\}}{\text{argmin}} \underset{\text{PC}}{\text{Score}}\left(K, \mathcal{U}\right)$
 2: $iter \leftarrow 0$
 3: while $iter < B_{pc}$ do
 4:     $\mathcal{B}^{iter} \leftarrow \text{BatchSampling}\left(\mathcal{U}, b\right)$
 5:     $\{\mathbf{x}_1^{iter}, \cdots, \mathbf{x}_k^{iter}\} \leftarrow \text{K-Means}\left(\mathcal{B}^{iter}, K_o\right)$
 6:     if $\{\mathbf{x}_1^{iter}, \cdots, \mathbf{x}_k^{iter}\} == \{\mathbf{x}_1^{iter-1}, \cdots, \mathbf{x}_k^{iter-1}\}$ then
 7:         Break
 8:     $iter \leftarrow iter + 1$
 9: $\mathcal{L} \leftarrow \{\left(\mathbf{x}_1, y_1\right), \cdots, \left(\mathbf{x}_k, y_k\right)\} \leftarrow \text{Labeling}(\{\mathbf{x}_1, \cdots, \mathbf{x}_k\})$
10: $\{\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_{\boldsymbol{T}}\} \leftarrow \text{Training}\left(f\left(\mathbf{x}, \boldsymbol{\theta}\right), y\right)$, where $\left(\mathbf{x}, y\right) \in \mathcal{L}$
11: $iter \leftarrow 0$
12: while $iter < B_{uu}$ do
13:     $\mathcal{B}^{iter} \leftarrow \text{BatchSampling}\left(\mathcal{U}, b\right)$
14:     $\mathcal{S} \leftarrow \emptyset$
15:     while $i < N_{uu}$ do
16:         $\mathbf{x}_i^* \leftarrow \text{argmax} \underset{\text{UU}}{\text{Score}}\left(\mathbf{x}\right),$ where $\mathbf{x} \in \mathcal{B}^{iter} \backslash \mathcal{S}$
17:         $\mathcal{S} \leftarrow \mathcal{S} \cup \mathbf{x}_i^*$
18:     $\mathcal{L} \leftarrow \mathcal{L} \cup \text{Labeling}\left(\mathcal{S}\right)$
19:     $\mathcal{U} \leftarrow \mathcal{U} \backslash \mathcal{S}$
20:     $\{\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_{\boldsymbol{T}}\} \leftarrow \text{Training}\left(f\left(\mathbf{x}, \boldsymbol{\theta}\right)\right)$, where $\mathbf{x} \in \mathcal{L}$
21:     if $\mathcal{U} == \emptyset$ then
22:         Break
23:     $iter \leftarrow iter + 1$
```

we select the top n candidate instances with the highest scores $\mathcal{S} = \{\mathbf{x}_1^*, \cdots, \mathbf{x}_n^*\}$ where $n \leq b$. This problem can be formulated as follows:

$$\mathbf{x}_i^* = \underset{\mathbf{x} \in \mathcal{B} \backslash \{\mathbf{x}_j^* | j < i\}}{\text{argmax}} \underset{\text{UU}}{\text{Score}}\left(\mathbf{x}\right) \tag{9}$$

The UUPC algorithm is shown in Alg. 1. In line 1 of Alg. 1, we select the optimized number of clusters K_o using Eq. 2. In lines 2-8, we choose batches randomly to train the mini-batch-based K-Means model until the positions of cluster centers are not changed. In line 9, the instances, which are the closest to the cluster centers, will be annotated by the oracle and moved into $\mathcal{L}$. Line 10 means training the BNN based on $\mathcal{L}$ to help the model understand the data space at initial AL phase. In lines 11-23, we calculate $\underset{\text{UU}}{\text{Score}}$ (see Eq. 8) on the batches data iteratively and annotate the top N_{uu} instances per batch. The annotated instances are moved to $\mathcal{L}$ to update the BNN model. We stop the process when the budget is exhausted.



(a) λ=0.05, optimal $K_o = 5$ (b) λ=0.005, optimal $K_o = 50$

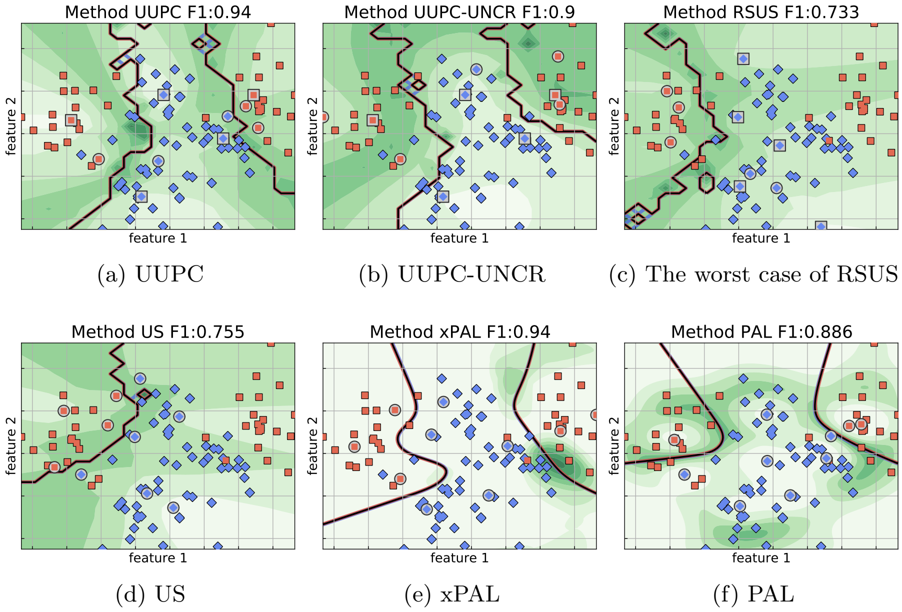
Fig. 1: Unsupervised cluster number evaluation by $\underset{\text{PC}}{\text{Score}}$ on artificial data set and MNIST using Eqs. 1 and 2. The four weights $\alpha_{1,\cdots,4}$ of $\underset{\text{PC}}{\text{Score}}$ are set to the same value of 0.25. The optimal number of clusters K_o locates at the lowest value of $\underset{\text{PC}}{\text{Score}}$. The red vertical dash line indicates the position of the optimal K_o.

4 Experimental Evaluation

To evaluate the quantitative performance of UUPC, we conduct experiments on artificial and real-world data sets. The following selection algorithms are compared. Besides random selection (RS) from a batch of instances and uncertainty sampling with entropy (US), we also use **R**andom **S**ampling strategies at the initial AL phase before **U**ncertainty **S**ampling (RSUS). For UUPC, K_o instances are selected by pre-clustering at initial AL phase. In order to make a fair comparison between our approach and RSUS, K_o instances are randomly selected as initial $\mathcal{L}$ in the RSUS method. To verify the utility metric, we conduct another strategy UUPC-UNCR, where only the uncertainty is considered, to assess the importance of the utility metric. For UUPC, we set the weights empirically in the $\underset{\text{UU}}{\text{Score}}$ as $\gamma_1 = 1.0$ and $\gamma_2 = 0.7$. In these experiments, we use a simple Bayesian dropout approximation neural network with multiple fully connected layers: 2 dense hidden layers with 250 and 100 units, ReLU activation and dropout, and an output layer. The dropout probabilities are set to 0.3 and 0.5 respectively. The $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_T$ are sampled ten times to obtain the average probability distribution on the label for each candidate instance, i.e., T is set to 10 in Eq. 4.

4.1 Artificial Data Set

The first experiment is inspired by [17]. Based on a low dimensional and small artificial data set that could visually show the acquisition behavior of different



(a) UUPC

(b) UUPC-UNCR

(c) The worst case of RSUS

(d) US

(e) xPAL

(f) PAL

Fig. 2: Visualization of acquisition behavior for different selection strategies on artificial data set. The green color indicates how useful a selection strategy considers a region. Darker areas are considered more valuable than brighter areas. The corresponding selection strategy has selected ten labeled instances marked with gray circles or squares. For UUPC, UUPC-UNCR and RSUS, the first selected five instances at AL initial phase are marked as gray squares. Thereby, one can see the current decision boundaries illustrated by black lines and how the usefulness is spatially distributed to select the next instance for labeling. The artificial data generation and visualization method are inspired by [17].

selection strategies. Through visualization, the performance of UUPC could be visually verified when it utilizes pre-clustering in the initial stage of AL and later selects samples through $\underset{\text{UU}}{\text{Score}}$. We also use F1 scores to quantitatively check whether our proposed method can outperform other baseline methods.

The artificial imbalanced data set contains 100 two-dimensional instances with two classes (60 blue diamonds and 40 red rectangles). We put the whole artificial data set as one batch and select an instance with most information from $\mathcal{U}$ at each AL iteration. One side the data size is too tiny another side it can compare with other traditional AL algorithms. Fig. 2d shows that US only has one unilateral decision boundary on the left side, which lacks exploration. The result of RS is not shown in Fig. 2 because it is unstable and entirely depends on random seeds. The optimized number of pre-clustering K_o is 5 (see Fig. 1a). For UUPC, UUPC-UNCR and RSUS, the initial selected five instances marked as gray squares are distributed in Figs 2a, 2b, and 2c respectively. In the UUPC and UUPC-UNCR methods, the initially selected instances are located in the five cluster centers representing the whole data space. Selecting the most representative instances could help the BNN model to estimate reliable uncertainty. However, similar to RS, the initial random selection strategy in RSUS relies on random seeds. Fig. 2c illustrates one of the worst cases of RSUS because all initial randomly selected instances belong to the blue diamonds class. This results in a poor quality model leading to poorer acquisition. The results in Figs. 2a and 2b prove that UUPC could increase the F1 score by 4% compares with UUPC-UNCR. Furthermore, we compare the results of the other two methods xPAL [17] and PAL [18] visually[1]. Since these two methods did not use BNN as a classifier, it might affect the output of their selection strategy. Here we only simply show the distribution of the labeled points. As shown in Fig 2e, our method could get similar F1 score as xPAL.

4.2 Real Data Set: MNIST

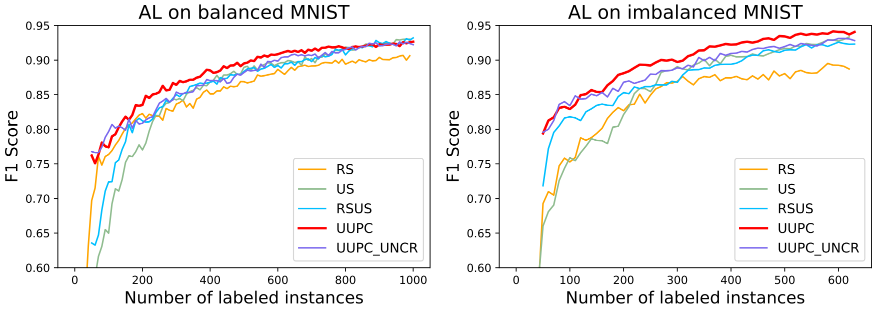
In the above experiment, we visualize the behavior of different selection strategies on low-dimensional artificial data. This experiment aims at evaluating the UUPC performance on real-world balanced and imbalanced data sets with high dimensions. MNIST [2] [19] data set includes 10 handwritten digits. The data set contains 20,000 training images and 10,000 testing images with the shape 28×28. As shown in Fig 1b, the hyperparameter of pre-clustering K_o is 50. The batch size is 1000, and we select the top 10 highest-scoring instances from each batch. We repeat the following experiments 20 times and evaluate the performance of different methods on the test data sets through the F1 score.

Fig. 3a illustrates the F1 scores of the test set with the different amount of annotations on balanced data set. We set the whole training set of MNIST to $\mathcal{U}$ and label 5% (1000 annotations) unlabeled instances in $\mathcal{U}$. Same as what [3, 17, 24] pointed out, US does even worse than RS when the number of annotations is smaller than 200 due to unreliable uncertainty estimations. UUPC and UUPC-UNCR outperform other methods at the initial phase because of pre-clustering. Due to only uncertainty is considered in UUPC-UNCR, the advantage of pre-clustering decays gradually after 200 annotations. When the number of labeled instances exceeds 250, the F1 score of RS increases slower than UUPC and US. It indicates that the uncertainty estimation given by BNN gets more and more important once sufficient annotations are available. UUPC could keep a higher F1 score, which is up to 4.5% higher than other baseline methods, until the number of the annotated samples is greater than 800.

Imbalanced data sets are very common in real-world applications. As a preliminary experiment, we randomly drop 75% of samples of digits 5, 6, 7, 8, 9 in the training and test set, to assess the performance of the methods in

[1] The algorithms of xPAL and PAL, as well as visualization presented in Fig. 2, are implemented by Kottke et al. https://github.com/dakot/probal.

[2] Obtained from https://colab.research.google.com.



(a) $\mathcal{U}$ is balanced MNIST.

(b) $\mathcal{U}$ is imbalanced MNIST.

Fig. 3: Learning curves for MNIST data sets. Each plot shows the multi-class F1 score of UUPC and the competing algorithms on the test images after annotating up to 5% instances from the balanced or imbalanced MNIST unlabeled data set. The learning curve that reaches a high F1 score fast is considered best.

imbalanced data set. Similar to the experiment in the balanced data set, 5% (630 annotations) unlabeled instances in $\mathcal{U}$ will be annotated. Since UUPC and UUPC-UNCR use pre-clustering, the F1 score in the initial phase is still higher than other methods presented in Fig. 3b. Due to selected instances are less representative, and most of them belong to majority classes, the F1 score of RS almost stops increasing after 300 annotations. It is worth noting that when the number of annotations is less than 150, the F1 score of UUPC-UNCR is slightly higher than that of UUPC, which means that when there are fewer annotations, the utility criterion may introduce uninformative samples. One solution is to set utility weight γ_2 to 0 at the initial stage and increase its value corresponding to the number of annotations dynamically. When the size of $\mathcal{L}$ is greater than 150, the utility could enhance the exploration of the selection strategy and increase classification accuracy significantly. Compared with other methods, the F1 score of UUPC is 4.3% higher than other methods on average under the same amount of labeling. In other words, our proposed method reduces the annotation cost by 33.1% on average but achieves the same performance as other baseline methods.

5 Conclusion & Future Work

The direct use of US in DAL could face two main challenges: the unreliable uncertainty estimation in the initial AL phase leads to poor acquisitions and further results in a poorer model, and the lack of exploration of US leads to insufficient diversity of samples. In this article, we propose an effective DAL algorithm UUPC, which enables the model to explore the data space at the initial stage using pre-clustering, and enhance the exploration of uncertainty sampling continually based on a combination of uncertainty and utility metrics. The method is assessed in preliminary experiments. The experimental results show that our method outperforms the baseline methods in balanced and imbalanced data sets under few annotations.

This work can be further researched in these directions: (1) In the current preliminary experiment, we only apply a tiny three-layer linear network and flatten the image data without considering image features. Gal et al. [7] proved that CNN could improve the recognition accuracy under the same number of annotations. It is necessary to extract features through CNN from high-dimensional data in future experiments. (2) The batch-based K-Means algorithm is applied in pre-clustering to improve computational efficiency. It is worth using Autoencoder with CNN layer to reduce dimensionality and extract the most informative features before clustering in further research. (3) At present, we only do preliminary experiments on artificial and MNIST data sets to verify our proposed method's feasibility. Further evaluations are needed on more data sets in the future. Besides, we will compare other existing selection strategies in DAL in further research. (4) UUPC and others mentioned methods above might fail in anomaly detection scenarios. One potential solution is performing isolation forest [20] or DBSCAN [5] at the initial stage of AL to get the rough decision boundary and then refining the result through uncertainty-utility (UU) strategy. (5) The method of obtaining the optimal number of clusters proposed in Subsection 3.1 is still a heuristic algorithm. In different application scenarios, estimating the weights of each sub-score and regularization weight λ in Eq. 1 relies on expert experience. The Bayesian information criterion (BIC) and the Akaike information criterion (AIC) could also determine the appropriate number of clusters [28, 8]. The advantage is that they originally contain regularization and do not require experts to set additional weights.

Acknowledgments

This work is supported within the Digital-Twin-Solar (03EI6024E) project, funded by BMWi: Deutsches Bundesministerium für Wirtschaft und Energie/German Federal Ministry for Economic Affairs and Energy. Special thanks to Daniel Kottke. His paper [17] and corresponding code allow us to understand the details of different AL algorithms. Besides, we compared our proposed method with other AL algorithms based on his visualization code to verify our preliminary idea. Thanks to the colleagues from IES at the University of Kassel, whose reviews and comments have helped improve the manuscript.

References

1. Asghar, N., Poupart, P., Jiang, X., Li, H.: Deep active learning for dialogue generation. arXiv preprint arXiv:1612.03929 (2016)
2. Ash, J.T., Zhang, C., Krishnamurthy, A., Langford, J., Agarwal, A.: Deep batch active learning by diverse, uncertain gradient lower bounds. arXiv preprint arXiv:1906.03671 (2019)

3. Attenberg, J., Provost, F.: Inactive learning? difficulties employing active learning in practice. ACM SIGKDD Explorations Newsletter **12**(2), 36–41 (2011)
4. Caliński, T., Harabasz, J.: A dendrite method for cluster analysis. Communications in Statistics-theory and Methods **3**(1), 1–27 (1974)
5. Ester, M., Kriegel, H.P., Sander, J., Xu, X., et al.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Kdd. vol. 96, pp. 226–231 (1996)
6. Gal, Y., Ghahramani, Z.: Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In: international conference on machine learning. pp. 1050–1059. PMLR (2016)
7. Gal, Y., Islam, R., Ghahramani, Z.: Deep Bayesian active learning with image data. In: Precup, D., Teh, Y.W. (eds.) Proceedings of the 34th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 70, pp. 1183–1192. PMLR (06–11 Aug 2017), http://proceedings.mlr.press/v70/gal17a.html
8. Grall-Maes, E., Dao, D.T.: Assessing the number of clusters in a mixture model with side-information. In: ICPRAM. pp. 41–47 (2016)
9. Halkidi, M., Batistakis, Y., Vazirgiannis, M.: On clustering validation techniques. Journal of intelligent information systems **17**(2), 107–145 (2001)
10. He, T., Jin, X., Ding, G., Yi, L., Yan, C.: Towards better uncertainty sampling: Active learning with multiple views for deep convolutional neural network. In: 2019 IEEE International Conference on Multimedia and Expo (ICME). pp. 1360–1365. IEEE (2019)
11. Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.R.: Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580 (2012)
12. Houlsby, N., Huszár, F., Ghahramani, Z., Lengyel, M.: Bayesian active learning for classification and preference learning. arXiv preprint arXiv:1112.5745 (2011)
13. Huang, S.J., Jin, R., Zhou, Z.H.: Active learning by querying informative and informative examples. IEEE Transactions on Pattern Analysis and Machine Intelligence **36**(10), 1936–1949 (2014)
14. Janz, D., van der Westhuizen, J., Hernández-Lobato, J.M.: Actively learning what makes a discrete sequence valid. arXiv preprint arXiv:1708.04465 (2017)
15. Joshi, A.J., Porikli, F., Papanikolopoulos, N.: Multi-class batch-mode active learning for image classification. In: 2010 IEEE international conference on robotics and automation. pp. 1873–1878. IEEE (2010)
16. Ketchen, D.J., Shook, C.L.: The application of cluster analysis in strategic management research: an analysis and critique. Strategic management journal **17**(6), 441–458 (1996)
17. Kottke, D., Herde, M., Sandrock, C., Huseljic, D., Krempl, G., Sick, B.: Toward optimal probabilistic active learning using a bayesian approach. Machine Learning pp. 1–33 (2021)
18. Kottke, D., Krempl, G., Lang, D., Teschner, J., Spiliopoulou, M.: Multi-class probabilistic active learning. In: Proceedings of the Twenty-second European Conference on Artificial Intelligence. pp. 586–594 (2016)
19. LeCun, Y., Cortes, C.: MNIST handwritten digit database (2010), http://yann.lecun.com/exdb/mnist/
20. Liu, F.T., Ting, K.M., Zhou, Z.H.: Isolation forest. In: 2008 Eighth IEEE International Conference on Data Mining. pp. 413–422 (2008). https://doi.org/10.1109/ICDM.2008.17

21. Nguyen, H.T., Smeulders, A.: Active learning using pre-clustering. In: Proceedings of the twenty-first international conference on Machine learning. p. 79 (2004)
22. Ostapuk, N., Yang, J., Cudré-Mauroux, P.: Activelink: deep active learning for link prediction in knowledge graphs. In: The World Wide Web Conference. pp. 1398–1408 (2019)
23. Ranganathan, H., Venkateswara, H., Chakraborty, S., Panchanathan, S.: Deep active learning for image classification. In: 2017 IEEE International Conference on Image Processing (ICIP). pp. 3934–3938. IEEE (2017)
24. Ren, P., Xiao, Y., Chang, X., Huang, P.Y., Li, Z., Chen, X., Wang, X.: A survey of deep active learning. arXiv preprint arXiv:2009.00236 (2020)
25. Rousseeuw, P.J.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of computational and applied mathematics **20**, 53–65 (1987)
26. Sculley, D.: Web-scale k-means clustering. In: Proceedings of the 19th international conference on World wide web. pp. 1177–1178 (2010)
27. Sener, O., Savarese, S.: Active learning for convolutional neural networks: A core-set approach. arXiv preprint arXiv:1708.00489 (2017)
28. Teklehaymanot, F.K., Muma, M., Zoubir, A.M.: Bayesian cluster enumeration criterion for unsupervised learning. IEEE Transactions on Signal Processing **66**(20), 5392–5406 (2018)
29. Tong, S.: Active learning: theory and applications. Stanford University (2001)
30. Wang, D., Shang, Y.: A new active labeling method for deep learning. In: 2014 International joint conference on neural networks (IJCNN). pp. 112–119. IEEE (2014)
31. Wolf, G.W.: Facility location: concepts, models, algorithms and case studies. series: Contributions to management science: edited by zanjirani farahani, reza and hekmatfar, masoud, heidelberg, germany, physica-verlag, 2009, 549 pp. (2011)
32. Yin, C., Qian, B., Cao, S., Li, X., Wei, J., Zheng, Q., Davidson, I.: Deep similarity-based batch mode active learning with exploration-exploitation. In: 2017 IEEE International Conference on Data Mining (ICDM). pp. 575–584. IEEE (2017)
33. Zhdanov, F.: Diverse mini-batch active learning. arXiv preprint arXiv:1901.05954 (2019)
34. Zhdanov, F.: Diverse mini-batch active learning. arXiv preprint arXiv:1901.05954 (2019)