Egyptian Informatics Journal 24 (2023) 100412

Contents lists available at ScienceDirect

Egyptian Informatics Journal

journal homepage: www.sciencedirect.com

Adaptive batch mode active learning with deep similarity☆

Kaiyuan Zhang [a,*], Buyue Qian [b], Jishang Wei [c], Changchang Yin [a], Shilei Cao [a], Xiaoyu Li [a], Yanjun Cao [d], Qinghua Zheng [a]

[a] School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an, Shaanxi 710049, China
[b] Beijing Chaoyang Hospital, Capital Medical University, BeiJing 100020, China
[c] HP Labs, 1501 Page Mill Rd, Palo Alto, CA 94304, USA
[d] Biomedicine Key Laboratory of Shaanxi Province, Northwest University, Xi'an 710071, China

ARTICLE INFO

Keywords:
Active learning
Adaptive batch mode active learning
Classification model
Deep neural network
Deep learning

ABSTRACT

Active learning is usually used in scenarios where few labels are available and manual labeling is expensive. To improve model performance, it is necessary to find the most valuable instance among all instances and label it to maximize the benefits of labeling. In practical scenarios, it is often more efficient to query a group of instances instead of a individual instance during each iteration. To achieve this goal, we need to explore the similarities between instances to ensure the informativeness and diversity. Many ad-hoc algorithms are proposed for batch mode active learning, and there are generally two major issues. One is that similarity measurement among instances often only relies on the expression of features but it is not well integrated with the classification algorithm model. This will cut down the precise measurement of diversity. The other is that in order to explore the decision boundary, these algorithms often choose the instance near the boundary. It is difficult to get the true boundary when there are few labeled instances. As a large number of instances continue to be labeled, information between instances is less used, and the performance will be greatly improved if it is properly used. In our work, we propose an adaptive algorithm based on deep neural networks to solve the two problems mentioned above. During the training phase, we established a paired network to improve the accuracy of the classification model, and the network can project the instance to a new feature space for more accurate similarity measurement. When batch labeling instances, we use the adaptive algorithm to select the instance by balancing the maximum uncertainty (exploration) and diversity (exploitation). Our algorithm has been validated for heart failure prediction tasks in real-world EHR datasets. Due to the no public of EHR data, we also conducted validation on two other classic classification tasks. Our algorithm is superior to the baseline method in both accuracy and convergence rate.

1. Introduction

Owing to the hard obtain of supervised data, numerous active learning algorithms have referred to select the most meaningful data from a collection of unlabeled data instances. Active learning algorithm typically selects an instance at one time to query an Oracle, and then repeats this procedure until the accuracy of the algorithm or the labeling budget is achieved. The strategy of selecting instances is crucial for algorithms with limited budgets. A good query strategy exposes the most informative data instances, which is expected to improve the model performance if they get labeled. There are many frameworks for active learning, such as exploring structure, minimizing expected error and variance, uncertainty sampling, searching hypothesis space.

In traditional active learning that relies on deep neural networks, the model needs to be frequently trained when there is even slight change in the training data. This is very inefficient and could lead to severe overfitting. Therefore, we use active learning with batch mode, that is, picking multiple instances at once to avoid this problem. The most

☆ This work is sponsored by "The Fundamental Theory and Applications of Big Data with Knowledge Engineering" under the National Key Research and Development Program of China with grant number 2016YFB1000903, and Project of China Knowledge Centre for Engineering Science and Technology, and Innovative Research Group of the National Natural Science Foundation of China 6172100, and Innovation Research Team of Ministry of Education IRT_17R86, and Project of China Knowledge Centre for Engineering Science and Technology.

* Corresponding author.
E-mail address: sxzky@stu.xjtu.edu.cn (K. Zhang).

https://doi.org/10.1016/j.eij.2023.100412
Received 19 July 2023; Received in revised form 9 September 2023; Accepted 14 October 2023
Available online 4 November 2023

common problem encountered when selecting a batch of instances is that the selected instance is informative but homogeneous. Since the detail provided by similar instances is almost identical to the learning model, the Oracle provided by expert is wasted. Hence diversity is the key to active learning in addition to informative batch processing mode.

A series of heuristic algorithms for BMAL has been proposed which would select informative and diverse instances. Some methods [1–4] first cluster unsupervised instances and then select instances from different classes to reduce redundancy between selected instances. The SVMactive algorithm [2,5–8] chooses a batch of uncertain and varied instances by defining a similarity function in projected kernel space.

However, the existing BMAL algorithms are mainly limited to two problems: first, the measurement of similarity between instances has a significant impact on the performance of the BMAL algorithm. For example, SVMactive maps instances to the kernel space, and then simply uses eigenvectors to calculate the similarity between instances. This is less diverse than the algorithm that learns with the classification model. Secondly, the existing BMAL algorithm is concentrating on "exploitation". When the learner chooses an instance, it often uses the existing hypothesis distribution to select the next instance. In the early stage of the learner, the instances that have been labeled are relatively insufficient. At this time, "exploration" is very important. Exploration can search for new areas which may be at the decision boundary. As an illustration, Osugi and Kun mention an exclusivity or problem that the upper left and lower right areas are negative, while the upper right and lower left areas are positive in reference [19]. If the classifier only marks the upper left, upper right, and lower left regions during the learning process, the instances in the lower right region will not be marked negative and the hypothesis space will not be adjusted.

Our research aims to design an effective batch mode active learning algorithm, aiming to enhance the performance of classifiers in situations where the amount of labeled data is limited and the cost of labels is high. The problem we need to address is how to select a fixed number of instances to maximize overall information while minimizing redundancy proactively. We need to overcome the two key challenges mentioned above. Firstly, we need to find a better way to measure similarity. Secondly, based on this measure of similarity, we need to develop a strategy that can select instances that are both informative and diverse. We propose an adaptive BMAL method that uses deep neural networks to learn similarity and balance exploration and exploitation. Exploration strategy represents gaining more information, while exploitation strategy means reducing redundancy.

In our method, convolutional neural network is used to classify the instance. The outcome of the last but one layer of the network can be considered as the learned instance feature representation. The inner product of feature vectors can be used to obtain the similarity between instances. Our neural network training process has two objectives: 1) improve the accuracy of training models, 2) improve the accuracy of similarity measurement between instances by mapping instances to a new space.

We evaluated our method on three data sets. The medical data set is the first data set (Electronic Health Records data warehouse), which is a real-world dataset used to determine whether patients will experience heart failure for a short period of time in the future. The next is the MNIST dataset, which is a commonly used handwritten digital image dataset in machine learning. It is a dataset of 60,000 handwritten digital images, containing image information and label information. Each image (28 * 28 pixels) can be expanded into a 784 length one-dimensional vector. The third dataset is the Fashion MNIST dataset, which contains 10 categories of images, which better reflects the algorithm's capabilities than the MNIST dataset. This experiment constructs a five-layer network structure to recognize 70,000 Gy-scale images of 10 categories. Experimental results show the effectiveness of our algorithm.

To sum up, our work has made the following contributions:

1) Using the multi-objective nature of deep networks, we propose an effective adaptive batch mode active learning algorithm.
2) We map the features of instances to a new representation through deep neural network learning so that we can better measure the similarity among instances.
3) The adaptive Exploration-Exploitation strategy that we propose allows us to approximate the distribution of real data more accurately, enabling us to choose instances that are both informative and diverse.
4) Our algorithm is effective enough when the labels are hard to obtain or the initial labels are few.

As far as we know, contributions 2 and 3 have not been explored in previous articles. It is the first attempt in BMAL. Differences between this article and previous conference versions. Based on the conference version, this paper does the following work which is not in the conference version [33]:

1) Add adaptive strategy to improve the accuracy and convergence speed of the algorithm.
2) Fine tune and refine the structure of the algorithm.
3) More quantitative methods to show data results.
4) Expand new experiments and discussions.

The rest of our paper is arranged as follows. Section II introduces related works. Mainly including the progress of relevant research on active learning, batch mode active learning, exploration–exploitation strategy and deep learning. In section III, we have elucidated the primary concept of the study method. Section IV shows our empirical research. We verified the effectiveness of each component of our algorithm on three different datasets. Section V summarizes our work.

2. Related work

2.1. Active learning

The purpose of active learning is to pick fewest instances with abundant information to build a high-performance model. Generally speaking, there are two types of active learning methods: representative sampling and uncertain sampling. Huang et al. [1] propose a representative sampling approach, QUIRE, to pick informative and representative instances. In [3–5], unlabeled instances are first clustered and then representative ones are chosen from each cluster. The property of these methods are heavily decided by the clustering algorithm [8] and [9] leverage a "query by committee" strategy to select the most informative instances. In order to solve manually creating labels problem, a unified multi-class active learning method is proposed by Yan et al. [2] to automatically label video data. This approach evaluates several practical sample selection strategies and advances active learning to handle multi-class problems. Chattopadhyay et al. [6] present a batch processing strategy that chooses a group of instances based on the distribution of unlabeled data. There are also some studies combine uncertainty sampling andrepresentative sampling [6,10]. The algorithm discussed above focuses on selecting the instance with the largest amount of information at each query. However, this is an inefficient way to tag instances one by one.

2.2. Batch mode active learning

In batch active learning, it is important to decrease redundancy when selecting instances. Hoi [11] picks a set of least redundant instances with minimal Fisher information. Joshi [12] proposed an active learning framework based on multi class image classification system. Schohn [13] proposes a greedy active learning algorithm based on SVM and applies it to document classification. Brinker [4] proposes a batch instance selection algorithm based on SVM in hyperplane of feature

space. Xia [5] uses a clustering algorithm to cluster unlabeled instances, and then selects instances from each cluster. Guo et al. [14] proposed an active learning algorithm for discriminating batch patterns, which defined the case selection task as a continuous optimization problem. Zhu et al. [15] In order to minimize the risk of harmonic energy minimization function, Gauss random field model is used for active learning and greedy query from unlabeled data. Shi [10] presents maximum impact, minimum redundancy and maximum uncertainty criteria to measure the informativeness of a collection of data. He transformed the BMAL problem into maximizing the objective function by selecting a collection of informative and diverse instances. Dasarathy [16] proposes a binary label prediction algorithm called S2 for active learning on a grap.

2.3. Exploration and exploitation

In the early days, the selected instances of the batch mode active learning algorithm may not cover all the distributions. Baram [17] presented an active learning algorithm to solve the problem of multi armed bandits using exploration and development strategies. Bondu [18] proposed an active learning strategy based on Bayesian form, which minimized the hypothesis of data and affected the exploration and development of input data space. Osugi [19] introduced an active learning algorithm to balance exploration and refine decision boundaries by dynamically tuning the exploration chance of every iteration. Cebron et al. [20] propose a new Prototype Based Active Learning (PBAC) approach with self-controlled exploration/exploitation strategy for classification. In [21–24], algorithms are proposed to discover new categories when all categories are unknown ahead of time. In recent years, many works have also emerged about soft computing strategies, such as Wang et al. [34] present a cooperative memetic algorithm with feedback (CMAF) and consider multiple selection strategies to solve the energy-aware distributed flow-shop with flexible assembly scheduling problem (EADFFASP). Pan et al. [35] propose a knowledge-based two-population optimization (KTPO) algorithm to handle the distributed energy-efficient parallel machines scheduling problem. For the carbon-efficient distributed blocking flow shop scheduling problem (CEDBFSP), Zhao et al. [36] propose a Pareto-based discrete Jaya algorithm (PDJaya) and demonstrate the effectiveness of the algorithm in solving CEDBFSP through numerical experiments. Based on Q-learning, Zhao et al. [37 38] propose a hyperheuristic with Q-learning (HHQL) to solve the energy-efficient DBFSP (EEDBFSP) and develop a cooperative meta-heuristic algorithm based on Q-learning (CMAQ) to address energy-efficient distributed no-wait flow-shop scheduling problem with sequence-dependent setup time. Zhao et al. [39] propose a population-based iterated greedy algorithm (PBIGA) to handle distributed assembly no-wait flow-shop scheduling problems. In general, there are two rules to follow when exploring. One is to find the most representative instances [20], and the other is to pick the instances most different from previously selected instances [19,25].

2.4. Deep learning

Deep neural networks have been successful in many areas, including object recognition, text classification, image classification and speech recognition etc. Many networks have been explored in the field of image recognition. Simonyan et al. [26] studied the influence of the depth of convolution network on the accuracy of large-scale image recognition settings. The work of [27] proposes a residual learning framework to simplify the training of networks deeper than those previously used. Deep neural network has also achieved excellent performance except for image recognition. Target detection has been a big driver in the work of [28 29]. Zhu et al. [30] proposed a novel model, which can be used to assess patient similarity using deep convolution neural network. Kim [31] proposes a convolutional neural networks for sentiment analysis and question classification. As the neural network becomes more and more complex, more training data are needed. But at the same time, a lot of labeled data are difficult to obtain. We aim to use active learning algorithm to reduce the workload of manual annotation and ensure the accuracy of the network.

3. The method

In this part, we propose an adaptive batch mode active learning algorithm based on deep neural networks. Due to the purpose of our algorithm being to utilize the adaptive exploration- exploration strategy to select and label informative and dissimilar instance, it can reduce the labeling workload in active settings.

3.1. Definition and settings

In this section, we first describe the problem as following. In pool based settings, the algorithm maintains a set of unlabeled instances and actively selects samples to be labeled by Oracle. Assuming there are u unlabeled data points $x_1, x_2,\ldots,x_u$ in the maintenance pool. There are l labeled instances (x_{u+1},y_{u+1}), (x_{u+2}, y_{u+2}), …, (x_{u+l}, y_{u+l}) in another set and l is much smaller than u. ~~L is much greater than u.~~ $\boldsymbol{L}=\{u+1,u+2,\ldots, u+l\}$ is the labeled pool when $\boldsymbol{U}=\{1, 2, \ldots, u\}$ is the unlabeled pool. The number of all instance collections is $n=u+l$. We represent the label of $x_1, x_2,\ldots,x_n$ as $y_1, y_2,\ldots,y_n$. b is the number of instances that can be labeled in an unlabeled pool. Our goal is to find a classifier h: X *to* Y that can label less than b instances while minimizing Generalization error $Err(h)$. X is the instance collection, and Y is the label space.

$$Err(h)=\frac{1}{2|L|}\sum_{(x,y)\in L}[1-equal(h(x),y)] \tag{1}$$

$$equal(a,b)=\begin{cases}1 & \text{if } a=b\\ -1 & \text{else}\end{cases}$$

The purpose of active learning is to boost the classifier effecting by marking as few instances as possible. The key to the problem is how to choose a batch of b instances to maximize classifier performance. The selected set of instances should be both diverse and informative. Each time the instance is selected and labeled, the classifier will retrain with the new labeled data instances set $\boldsymbol{L}$. The basic model is described in algorithm 1.

The method is to divide the budget into multiple batches and retrain the learners for each batch. There are two problems in this process that need to be solved. The first is how to select the set of instances with the most information, and the second is how to use labeled data when retraining learners. Due to the training of early labeled data in each iteration, the deep neural network classifier used in the experiment is prone to overfitting in small-scale labeled datasets. These will cause problems during training as illustrated in the following sections.

For a clear understanding of our model, the involved notations are

Table 1
Notations.

Symbol	Description
$\boldsymbol{U}$	Unmarked instance pool
$\boldsymbol{L}$	Marked instance set
$\boldsymbol{S}$	Chosen instance set
i	Index of data point
x_i	i-th data point
y_i	i-th data point's label
f_i	i-th data point feature vector
k	Number of data points to be selected for each batch
m	Number of data points marked in exploitation
b	Number of data points that can be labeled overall

summarized in Table 1.

Algorithm 1 Basic model

Input: h, $\boldsymbol{U}$, $\boldsymbol{L}$, b, k
Output: h
1: **for** $i = 0$ to $|\boldsymbol{L}| = b$ do
2: Select k instances $\boldsymbol{S}$;
3: $\boldsymbol{L} = \boldsymbol{L} + \boldsymbol{S}$;
4: $\boldsymbol{U} = \boldsymbol{U} - \boldsymbol{S}$;
5: retrain h with renewed $\boldsymbol{L}$;
6: **end for**

3.2. Exploitation

Two standards for calculating the quantity of instance information. One is the minimal redundancy and the other is maximum uncertainty as shown below.

1) Maximum uncertainty

The uncertainty sampling strategy focuses on instances that are easily misclassified, such as those close to the decision boundary. In general, the uncertainty sampling strategy uses the level of the entropy to determine the selected instance.

$$\mathrm{E}(x) = -\sum_{0<<i<<|Y|} h_i(x)\log(h_i(x)) \tag{2}$$

We can simply see that the function maximizes when the variable is located at the decision boundary. The probability that x belongs to the i-th class can be denoted as $h_i(x)$. E ($\boldsymbol{S}$) represents the sum of uncertainties for all data points in set $\boldsymbol{S}$, which is the entropy of set $\boldsymbol{S}$.

$$\mathrm{E}(S) = \sum_{i\in S}\mathrm{E}(x_i) \tag{3}$$

2) Minimum redundancy

Under our settings, the algorithm selects a group of instances to label each iteration. This means that the algorithm not only considers the maximum entropy value of instances, but also considers the diversity of selected instances. So the criteria for selecting redundancy of set $\boldsymbol{S}$ needs to be updated. We need to measure the similarity between instances in an appropriate way to select the different types of instances. Similarity is usually measured in feature space. Kernel space is often used in SVM. The feature space used by the algorithm is the result of the penultimate layer of the neural network, with softmax being the last layer. f_i is the feature vector of data point i. The similarity between instance i and instance j is defined as:

$$\mathrm{Sim}(i,j) = f_i M f_j \tag{4}$$

The similarity matrix M in the function can be learned in the network.

We define the total similarity of a set as $\boldsymbol{R}(\boldsymbol{S})$.

$$R(S) = \sum_{i\in S}\sum_{j\in S} Sim(i,j) \tag{5}$$

We define maximum uncertainty and minimum redundancy with two functions respectively. We use these two functions to naturally formulate our object function $\boldsymbol{I}(\boldsymbol{S})$.

$$I(S) = E(S) - \frac{\alpha}{|S|}R(S) \tag{6}$$

where $\alpha > 0$ is the weighting parameter used to balance uncertainty and redundancy.

Since finding an $\boldsymbol{S}$ to make $\boldsymbol{I}(\boldsymbol{S})$ maximal is very difficult and costly, we use greedy strategies to gradually find the optimal solution. The algorithm first selects an instance of maximum uncertainty and then selects an instance with the most uncertainty and minimum redundancy from the selected set. Our greedy algorithm is summarized in Algorithm 2. The similarity between a data point and a dataset is defined as:

$$\mathrm{Sim}(i,\boldsymbol{S}) = \max_{j\in\boldsymbol{S}}(\mathrm{Sim}(i,j)) \tag{7}$$

Our method chooses the next data point with the maximum $\boldsymbol{I}(i)$:

$$\boldsymbol{I}(i) = \boldsymbol{E}(x_i) - \alpha\mathrm{Sim}(i,\boldsymbol{S}) \tag{8}$$

3.3. Exploration

At the start, we think that instances being labelled do not cover all potential distributions because we have fewer instances to be labeled. At this time our algorithm is more focused on exploring unknown areas. So we choose the instance that is farthest from the marked group. Equation (7) to measure the similarity between a data point and a data set. Formally, we formulate our exploration object function as follows:

$$\boldsymbol{S} = \min\sum_{i\in\boldsymbol{S}}\mathrm{Sim}(i,\boldsymbol{L}\cup\boldsymbol{S}) + \sum_{i\in\boldsymbol{S}}\sum_{j\in\boldsymbol{S}}\mathrm{Sim}(i,j) \tag{9}$$

Solving Equation (9) is clearly an NP hard problem. We are trying to use a greedy strategy instead of solving Equation (9). We use Equation (10) to choose such an instance.

$$i = \min_i \mathrm{Sim}(i,\boldsymbol{L}\cup\boldsymbol{S}) \tag{10}$$

When we get an instance each time using Equation (10), the chosen set will merge this instance. We summarize this process in Algorithm 2.

Algorithm 2 Greedy adaptive collection

Input: a, δ, m, k, h, $\boldsymbol{L}$, $\boldsymbol{U}$
Output: chosen set $\boldsymbol{S}$
Initialize: Initialize the $\boldsymbol{S}$ set as an empty set
Compute entropy value $\boldsymbol{E}$;
Compute overall similarity **Sim**;
Find an instance i belonging to $\boldsymbol{U}$ that maximizes $\boldsymbol{E}(x_i)$;
$\boldsymbol{S}$ set Merge i *to* $\boldsymbol{S}$
 # self-adaption
for *index* = 1 to a **do**
  for $i \in \boldsymbol{U} - \boldsymbol{S}$ **do**
    $\boldsymbol{I}(i) = -\mathbf{Sim}(i, \boldsymbol{L}\cup\boldsymbol{S})$
  end for
  Find $i \in \boldsymbol{U} - \boldsymbol{S}$ to maximize $\boldsymbol{I}(i)$;
  $\boldsymbol{S} \leftarrow \boldsymbol{S}\cup\{i\}$;
end for
Compare each $h(i)$ with y_i, where $i \in \boldsymbol{S}$;
Obtain matching probability mp;
Rounding m= round[(δ - mp) * m]
 # exploration
for *index* = a+1 to m **do**
  for $i \in \boldsymbol{U} - \boldsymbol{S}$ **do**
    $\boldsymbol{I}(i) = -\mathbf{Sim}(i, \boldsymbol{L}\cup\boldsymbol{S})$
  end for
  Find $i \in \boldsymbol{U} - \boldsymbol{S}$ to maximize $\boldsymbol{I}(i)$;
  $\boldsymbol{S} \leftarrow \boldsymbol{S}\cup\{i\}$;
end for
# **exploitation**
for *index* = m+1 to k **do**
  for $i \in \boldsymbol{U} - \boldsymbol{S}$ **do**
    $\boldsymbol{I}(i) = \mathbf{E}(x_i) - \beta\mathbf{Sim}(i, \boldsymbol{S})$
  end for
  Find $i \in \boldsymbol{U} - \boldsymbol{S}$ to maximize $\boldsymbol{I}(i)$;
  $\boldsymbol{S} \leftarrow \boldsymbol{S}\cup\{i\}$;
end for

3.4. Self-adaption

We use adaptive strategies to combine exploitation and exploration in BMAL. We divide each iteration of Algorithm 2 into three parts: adaptive phase, exploration phase, and exploitation phase. At the beginning, we assume that there are very few labeled instances and they cannot represent the true distribution of all instances. Thus, the algorithm should focus more on exploring to discover new areas better. The exploration and exploitation strategies are executed respectively at each iteration. We denote m and k-m as the numbers of instances to be selected in exploration and exploitation respectfully, which control the two strategies' weights.

As the labeled instance set becomes larger, the possibility of redundancy between the selected instance and the labeled instance set increases. If the value of m is fixed, it is difficult to balance exploration and exploitation throughout the entire process of the algorithm. We expect to initialize a relatively large m value and gradually optimize it during the iteration process based on the performance of the exploration and exploitation strategies until the end of the iteration.

As shown in Algorithm 2, we introduce a new parameter a ($a \ll m < k$) to verify the performance of current learners in exploration and exploration. This means that we select a instances to test learners by exploration strategy. We can obtain the test result mp, which is matching probability. Based on the test result mp, we adjust the value of m adaptively.

We define the formula for adaptive adjustment m as:

$$m = round[(\boldsymbol{\delta} - mp)^{*}m] \tag{11}$$

In Equation (11), the larger $mp \in$ (0 %,100 %), the better it is to explore the true underlying structure of the data, and it is difficult to discover new regions. The smaller the mp, the more exploration is still needed to explore the real underlying structure. δ is an adjustment coefficient used to control the strategy of m adjustment. In our experiment, $\delta = 1.5$ and the adjustment range of m is 50 % to 150 %. When $mp = 50$ %, the value of m remains unchanged.

3.5. Pairwise neural network

Instance similarity is often defined in feature space. However, these features are usually not computed to calculate the similarity but to classify. In our article, we propose a pairwise network using feature space to calculate similarity. We use deep neural networks with strong learning capabilities to learn classification tasks. There are two training objectives of our deep network, one is to enhance learner's precision as in (1), and the other is to improve the similarity of paired networks in this feature space, which is expressed as follows:

$$\mathrm{Err}_2(h) = -\frac{\beta}{\boldsymbol{S}^2}\sum_{i\in\boldsymbol{S}}\sum_{j\in\boldsymbol{S}}\mathrm{sim}(i,j)\mathrm{equal}\left(y_i, y_j\right) \tag{12}$$

A weight parameter is used to balance (1) and (12). Our deep network architectures shown in Fig. 1 forecast the chance of cardiac failure. In Fig. 2, pairwise model is to learn the similarity of paired instances.

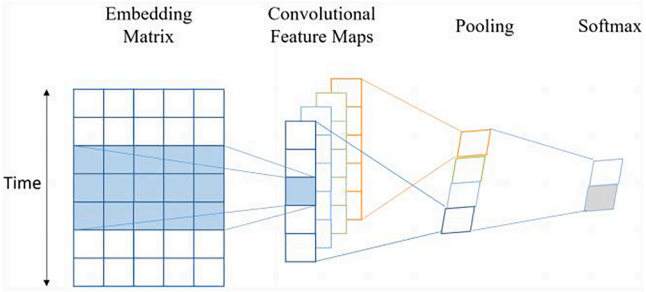


Fig. 1. The deep network architectures of heart failure prediction.

At the beginning of the training stage, we first learn the similarity between the instances, and then we use our active learning algorithm to select the instances that need to be labeled according to the similarity. In the test phase, we use the constructed network for reasoning.

4. Experimentation and evaluation

In this part, we assessed our proposed method (AEEP). This approach is an adaptive combination of exploitation and exploration. Our method is compared with these baseline:

- Batch mode active learning (BMAL), that picks a shipment of data points with the most informative and influence. This approach is equivalent to the exploitation component of our methodology outlined in Algorithm 2.
- Max entropy selection (EM), which picks a group of instances with the highest entropy value.
- Kernel farthest first (KFF), which picks instances in kernel space that is most different from the marked set from the unmarked set. This approach is equivalent to the exploration component of our methodology outlined in Algorithm 2.
- In passive random sampling (Random), a group of data points are randomly selected from pool to enhance learner.
- Batch mode active learning based on Exploration-Exploitation-Pairwise Model (EE), which is similar to our proposed method but without adaptive features.

For convenience, BMAL, EM, KFF, Random, EE are used to represent the baseline mentioned above respectively, and AEEP (Adaptive Exploration-Exploitation-Pairwise Model) to represent our method. These five baseline methods can represent the current approach to solving such problems in a comprehensive way. KFF is focused on exploration, so early on they performed much better than others. BMAL is more development-oriented, so it stands out in later stages. Entropy maximization method is also a very classic method to deal with problems. It sets an entropy value for each point to be amplified, and recalculates the entropy after output a point, and outputs the point with the maximum entropy. Besides, random selection is also chosen as a baseline method.

4.1. Heart failure prediction

In this experiment, we used a real clinical electronic health records database, which contains data from over 200 thousand people over 4 years. Electronic health records contain various medical data with time information, including basic information (such as gender, age, etc.), laboratory data, medication data, diagnostic data, surgical data, and so on. Heart failure is a common multifactorial condition that can lead to serious consequences if not detected in time.

To better study the early prediction of heart failure, we took a group of patients with heart failure as a case and a group of comparable individuals who did not have heart failure as a control, which is a case-control study.

For the purpose of forecasting heart failure at a later time, a standard date is required for case-control study. The day before the diagnosis of heart failure is the standard date for the case patients. The last day of the database is the standard date for the control patients. For all patient's EHR data, we defined six months (180 days) after the standard date as the prediction window and the previous as the observation window. Data during the observation window is used for analysis.

1) Patient's representation

To ensure the effective execution of the algorithm, we require at least 50 medical records during the observation window period when selecting patients. Our experts selected 2323 cases and 2323 controls on this standard, totaling 4626 patients. Due to the inconsistent number of medical events among patients, the sequence length of patients is also inconsistent. For the convenience of computation, we fill the sequence to

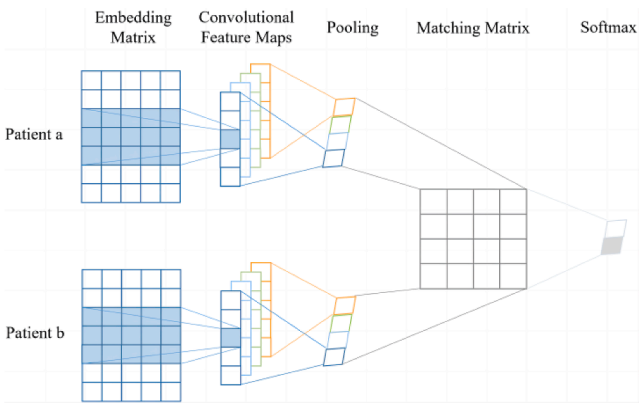


Fig. 2. The pairwise deep network architecture that cannot only classify but also learn similarity between pairwise patients.

the same length, and the filled content is meaningless in medical terms. We use EHR data to train a word embedding model, which can convert medical events into fixed length vectors. Each patient can be represented as a fixed size matrix.

2) Prediction model

As shown in Fig. 1, we have constructed a 4-layer convolutional neural network. This convolutional network extracts the core features of patient matrix data through convolution and pooling, and generates fixed length vectors. The last softmax layer is used for prediction. As shown in Fig. 2, the pooling result can be used as feature vector for patients. By iteratively optimizing the objective functions Err (h) and Err_2 (h), the similarity between patients can be obtained.

3) Performance evaluation

In order to more clearly show the performance of our method and baseline on the prediction of heart failure, we divided the training data into 20 parts, and measured the performance of all the methods at the time of each part arrival. The experiment was repeated 20 times, and the average result is shown in Fig. 3, Fig. 4 and Fig. 5. We also summarized the results in Table 2.Fig. 6.

As shown in Fig. 3, the results of active learning algorithms are superior to those of passive learning (Random), suggesting that active learning algorithms can be effectively integrated with deep networks. Furthermore, our algorithm, AEEP, demonstrates superior performance and fastest convergence compared to EE algorithms which do not have adaptive strategies, indicating the advantages of our adaptive combination of exploitation and exploration over non-adaptive algorithms. In Fig. 4, the fluctuations of all algorithms are compared, showing that our algorithm not only outperforms others in terms of results, but also shows greater stability. In Fig. 5, we verified the effectiveness of the pairwise network. The results show that pairwise networks can provide better similarity metrics, allowing for the selection of more diverse and informative instances at an early stage.

4) Parameter settings

There are five parameters in the experiment, k, a, m, α and β, all of which have an impact on the performance of the algorithm. k represents the number of instances extracted from each batch for labeling. We search for k values and set them to 2, 4, 8, 16, 32, and 64. The results indicate that smaller k values lead to better performance, but also that more deep network training iterations lead to lower efficiency. When k is less than or equal to 16, there is no significant fluctuation in the algorithm's performance. However, when k exceeds 16, the performance significantly decreases. The value a represents the proportion of adaptive strategies. The larger the value of a, the more accurate the value of m after dynamic adjustment. However, this can result in a decrease in the proportion of exploitation-exploration strategies being executed, which can severely affect the performance of the algorithm. When $k = 16$, we search for the value of a and set it to 1, 2, 3, 4, 5, 6. Experiments have shown that the value of a between 2 and 4 performs better. m represents the ratio of exploitation and exploration strategies, which can be dynamically adjusted using adaptive strategies. In general, when initializing m, a larger value is used to increase the proportion of the exploration strategy. In this experiment, m was initialized as k. α balances uncertainty and redundancy in equation (8). β is used to balance equation (1) and equation (12). In this experiment, it was found through search that the best performance was achieved when $\alpha = 1.5$ and $\beta = 0.01$. When β is greater, more accurate similarity functions can be trained in early stage to find instances with more information and variety. When the similarity function is relatively accurate, a large β can prevent the fitting of the classifier.

4.2. MNIST classification

We evaluate our framework on MNIST dataset. The classic MNIST dataset contains a large number of handwritten digital images. For more than a decade, researchers from the fields of machine learning, machine vision, artificial intelligence, and deep learning have used this data set as a benchmark against which to measure algorithms.

We use a classic seven layer neural network to classify MNIST images. The neural network consists of three convolutional layers, two pooling layers, and two fully connected layers. The output of the network is a 10 dimensional vector, corresponding to 10 categories. The entire network uses ReLU as the activation function. Our active learning algorithm is applied in the model.

We ran our method and the baseline for 20 times. The average iteration result is shown in Fig. 7 and the stability of different methods is shown in Fig. 8. We also summarized the results in Table 3.

As can be seen from Fig. 7, passive learning performed better than active learning at the beginning because the instances selected through active learning were not balanced enough. Due to difficulties in adapting to changes at different stages, passive learning converges slowly and has a weak generalization ability. The performance of the entropy maximization (EM) algorithm is lower than our batch mode algorithm, indicating that removing redundant instances can improve the performance of the algorithm. The adaptive exploration–exploitation algorithm we proposed can be seen as a dynamic combination of BMAL and KFF algorithms. In the initial stage, in order to better explore the boundaries, our algorithm adopts a similar strategy to KFF, resulting in similar performance and outperforming other algorithms. With continuous iteration, our algorithm dynamically balances the BMAL and KFF

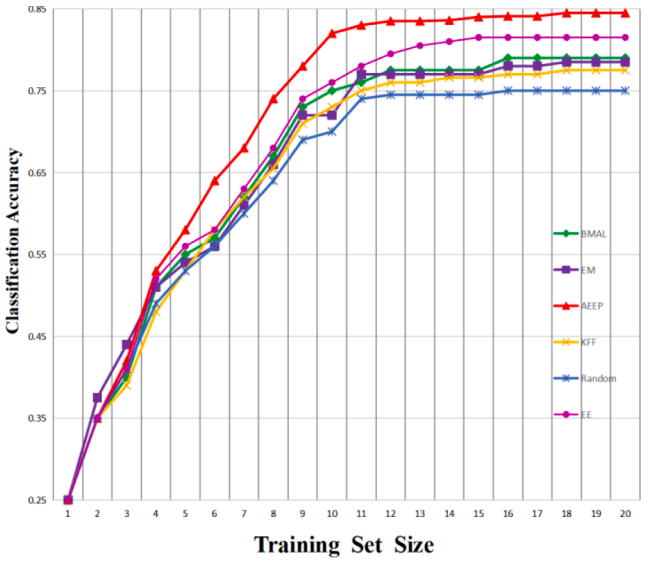
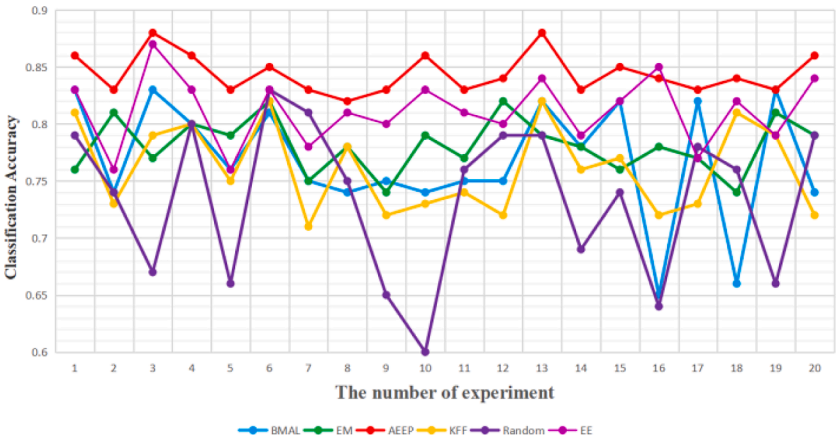


Fig. 3. The performance of our method and baseline on heart failure prediction.



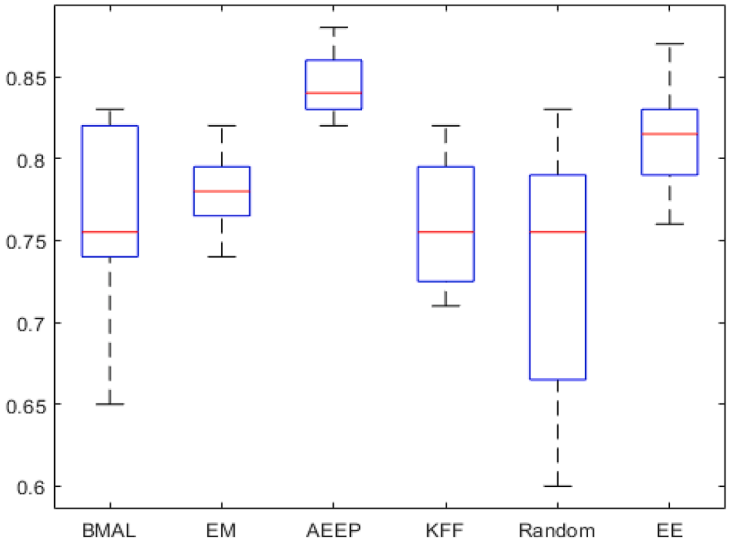


Fig. 4. Performance of running experiments 20 times.

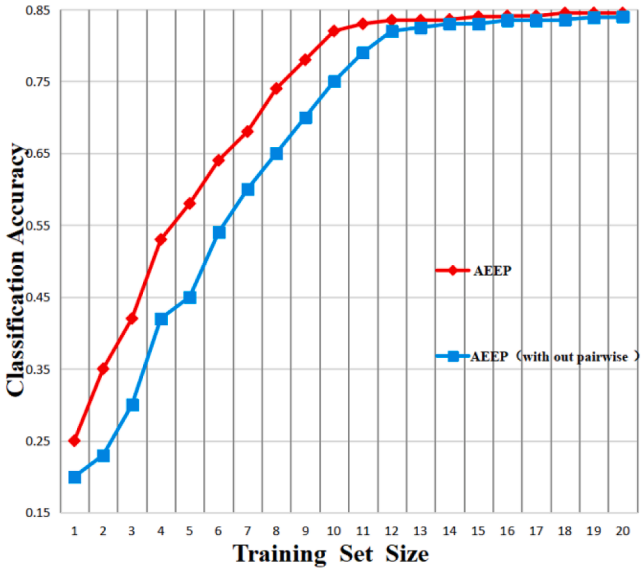


Fig. 5. AEEP with and without the pairwise model.

Table 2
Average performance of 20 experiments.

BMAL	EM	AEEP	KFF	Random	EE
0.79 ± 0.05	0.79 ± 0.02	0.85 ± 0.01	0.78 ± 0.04	0.75 ± 0.07	0.82 ± 0.03

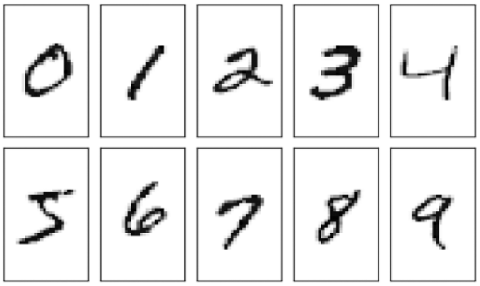


Fig. 6. Typical MNIST datasets, including a large number of handwritten numbers.

strategies. To reduce the similarity between selected instances, our algorithm gradually strengthens exploitation, thereby approaching the BMAL strategy. The findings suggest that exploration holds greater significance in the initial phase, whereas exploitation takes precedence

in the later stage. As evident from Fig. 8, our algorithm exhibits superior performance and stability.

Our deep network has two training objectives, one is to advance classification precision by satisfying equation (1). The other is to improve the similarity of the paired network in the feature space, as shown in equation (12). The weight parameter β is used to balance equation (1) and equation (12). Fig. 9 gives our algorithm's performance with various beta parameters. When β is greater, more accurate similarity functions can be trained in early stage to find instances with more information and variety. When the similarity function is relatively accurate, a large β can prevent the fitting of the classifier. In this experiment, it was found through search that the best performance was achieved when $\beta = 0.01$.

In the experiment, we set some noise in the data set to test the robustness of methods. When our methods selects and labels instances, there is a certain probability that the label will change. Fig. 10 gives our algorithm's performance with distinct noise rates, which demonstrate the robustness of our algorithm.

The training result of network is usually effected by batch size. Let the batch size in each iteration be k, which is set to 16, 32, 64 respectively. When batch size $k = 16$, except for random sampling, the performance of other methods decreases. BMAL, KFF, EM always focus on newly selected instances when the batch size is small, resulting in a set of selected instances that are usually highly unbalanced. The random algorithm is the least affected by the change of m, because the correlation coefficient between the selection of instances and the size of batch m is the least in random sampling. AEEP performs well due to its balanced selection of instances. When batch size $m = 32$, our AEEP and KFF method have excellent performance in the early stage, superior to other baseline methods. Since the BMAL method is more prone to exploitation, the results will be better in the later stages of algorithm execution. Fig. 7 shows how our method compares with other methods in batch size $m = 32$. When batch size $m = 64$, the performance of BMAL and KFF has been improved at the beginning, but the final performance is not as good as before. This is because with the increase of batch size, more instances can be labeled at the beginning, and the effect of network training will be enhanced. When these active methods run to the later stage, the amount of labeled instances is already very large. At this time, if you still focus on exploration, the performance will decline. Experimental results show that dynamically adjusting the proportion of exploration and utilization during algorithm execution can help improve performance.

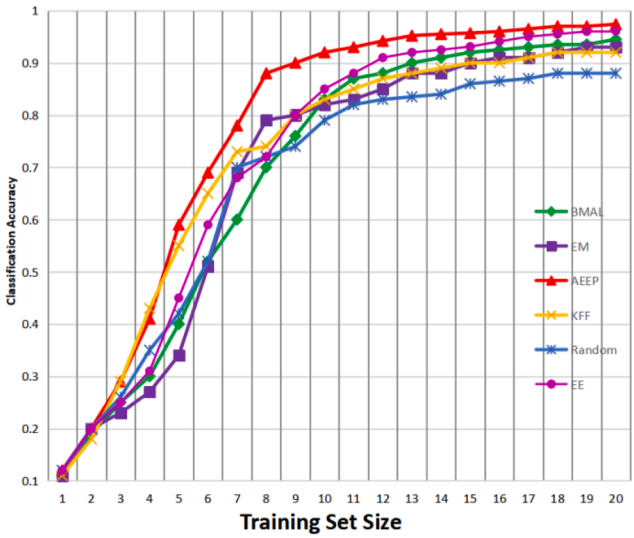


Fig. 7. The performance of our method and baseline on typical MNIST datasets.

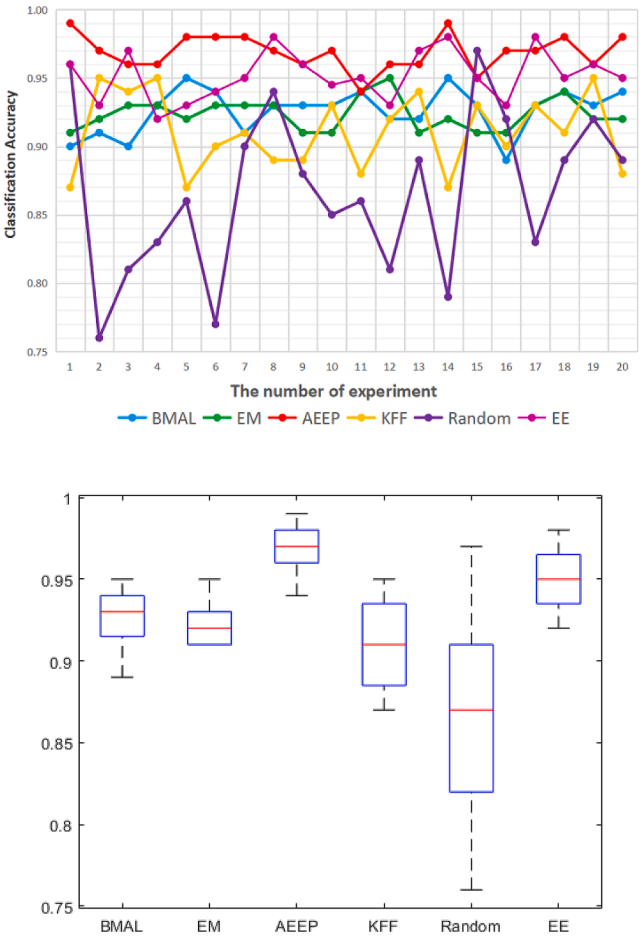


Fig. 8. Performance of running experiments 20 times.

Table 3
Average performance of 20 experiments.

BMAL	EM	AEEP	KFF	Random	EE
0.94 ± 0.02	0.93 ± 0.01	**0.97 ± 0.01**	0.92 ± 0.02	0.88 ± 0.06	0.96 ± 0.02

4.3. Fashion MNIST classification

The classic MNIST dataset has been used many times by different algorithms, and the results are good enough. It is difficult to show the superiority of the algorithm, so we choose fashion MNIST as the

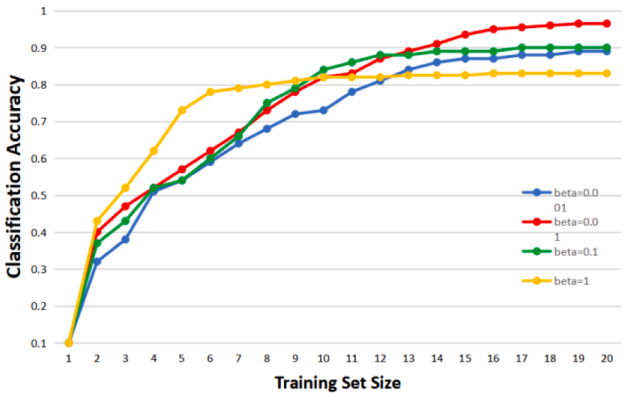


Fig. 9. Performance of our methods under different beta parameters.

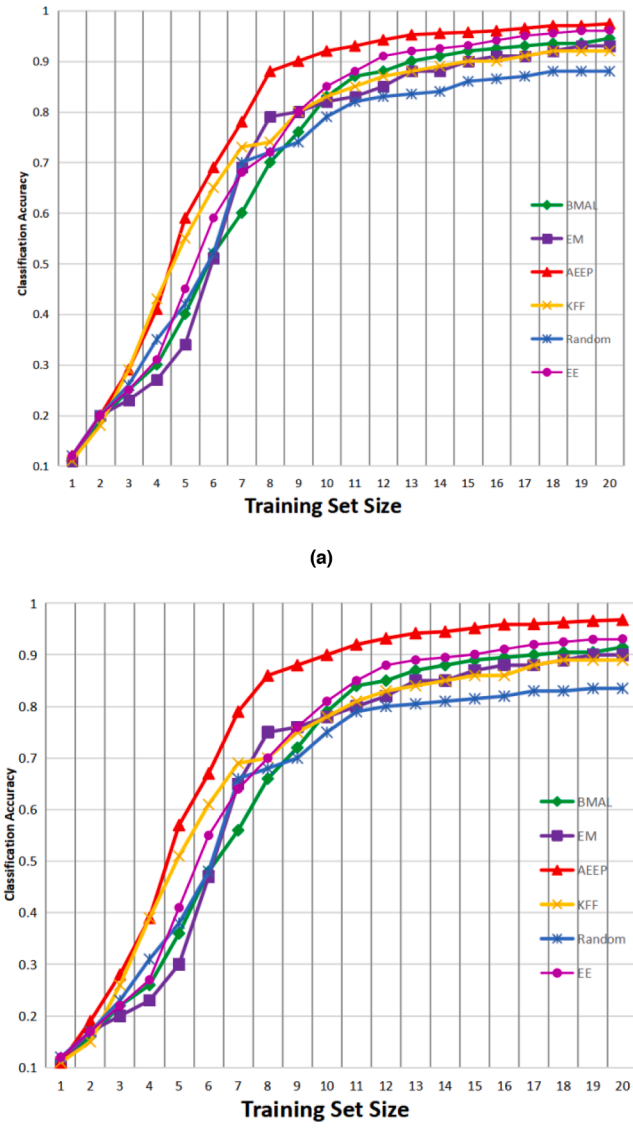


Fig. 10. MNIST classification results for (a) no noise, (b) noise rate eta = 0.15.

extended experiment. Compared with MNIST, fashion-MNIST is a much more complex image data set. It contains front images of 70,000 various fashion goods in 10 types, provided by Zalando. An example of fashion-MNIST how the data looks is shown in Fig. 11. Similar to the MNIST dataset, it divides training and test data according to 60,000 / 10000, and the gray scale of each picture is 28x28.

In Fashion MNIST datasets, β parameters still have a great impact on the results. Fig. 14 gives our AEEP algorithm's performance under different beta parameters.

Fashion MNIST is considered as an improved MNIST data set, which can verify the ability of algorithm in complex environment better than the original MNIST.

The experiment on the Fashion-MNIST is repeated for 20 times. The average of the results in each iteration of the algorithm is shown in Figs. 12, 13. The results are summarized in Table 4.

The results show that due to the effectiveness of our adaptive strategy, our algorithm has strong stability and performance in more complex datasets.

As can be seen from Fig. 12, in complex datasets, passive learning algorithms perform slightly better in the initial stage. This is because the data distribution chosen by active learning algorithms at the initial stage

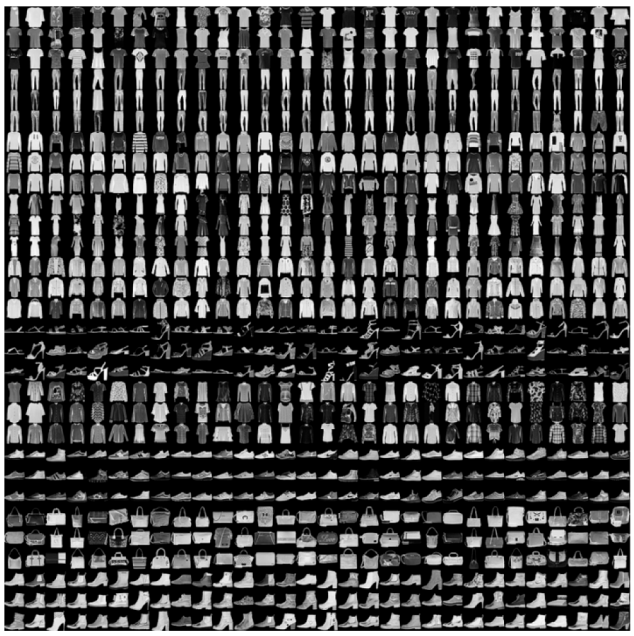


Fig. 11. An example of fashion-MNIST. Each class takes three-rows.

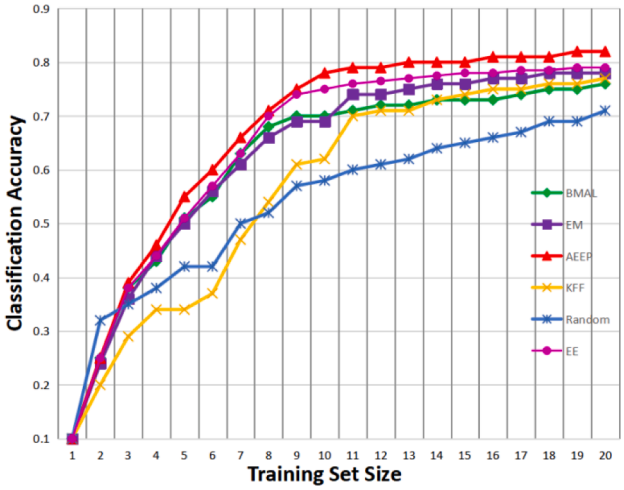


Fig. 12. The performance of our method and baseline on Fashion MNIST datasets.

differs significantly from the true distribution, while the data distribution chosen by passive learning algorithms is closer to the true distribution. However, with continuous iterative selection, the active learning algorithm converges faster and performs better.

5. Conclusion

In this paper, we address two common issues with batch mode active learning. We propose an adaptive batch mode active learning algorithm with deep similarity and successfully integrate it with deep neural networks. During training, we use pairwise deep neural network to extract the features of the instances and get the similarity between the instances. This enhances the diversity of selected data in active learning. In the active learning stage, we use an adaptive strategy to dynamically adjust the balance between maximum uncertainty (exploration) and diversity (exploitation) in order to select informative and diverse instances.

The experimental results indicate that pairwise networks can obtain

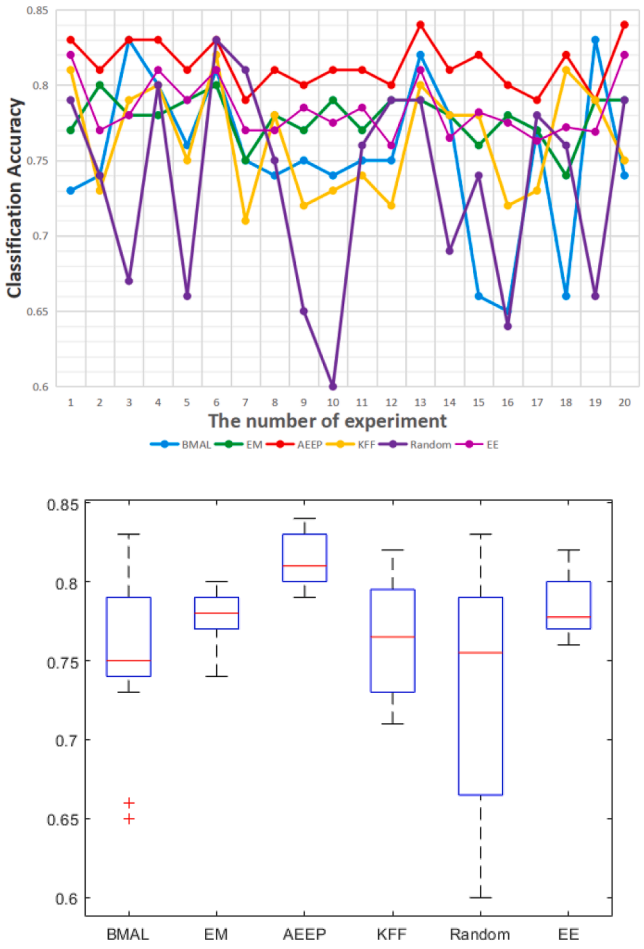


Fig. 13. Performance of running experiments 20 times.

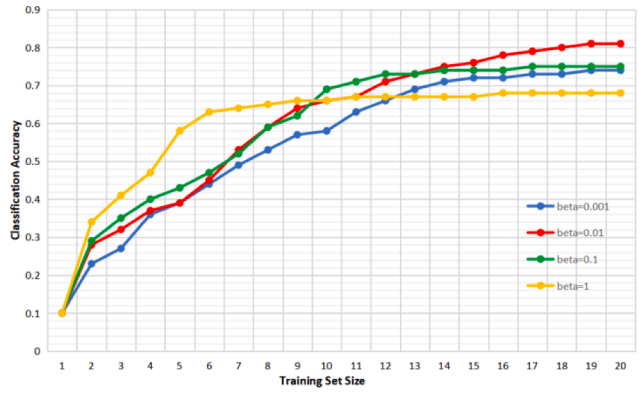


Fig. 14. Performance of our methods under different beta parameters on Fashion MNIST datasets.

better similarity metrics compared to single-mode networks, thereby enhancing the performance of our active learning algorithm. Using this accurate similarity measure, we propose an adaptive exploration–exploitation algorithm that can be viewed as a dynamic combination of BMAL and KFF algorithms. Experimental results show that our adaptive active learning algorithm outperforms the baseline significantly on multiple tasks.

In the future work, we plan to explore the difference between greedy solution and optimal solution, so as to find comprehensive guidelines to improve the performance of our method. Besides, we will combine matrix completion and data analysis in biology [32] to boost the convergence speed of our network in measuring similarity.

Table 4
Average performance of 20 experiments.

BMAL	EM	AEEP	KFF	Random	EE
0.75 ± 0.05	0.78 ± 0.02	**0.81 ± 0.02**	0.76 ± 0.04	0.74 ± 0.07	0.78 ± 0.02

CRediT authorship contribution statement

Kaiyuan Zhang: Conceptualization, Methodology, Software, Writing – original draft. **Buyue Qian:** Software, Validation. **Jishang Wei:** Resources, Investigation. **Changchang Yin:** Writing – original draft. **Shilei Cao:** Writing – review & editing. **Xiaoyu Li:** Visualization. **Yanjun Cao:** Funding acquisition, Writing – review & editing. **Qinghua Zheng:** Project administration, Supervision, Funding acquisition.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work is sponsored by "The Fundamental Theory and Applications of Big Data with Knowledge Engineering" under the National Key Research and Development Program of China with grant number 2016YFB1000903, and Project of China Knowledge Centre for Engineering Science and Technology, and Innovative Research Group of the National Natural Science Foundation of China 6172100, and Innovation Research Team of Ministry of Education IRT_17R86, and Project of China Knowledge Centre for Engineering Science and Technology.

References

[1] Huang SJ, Jin R, Zhou ZH. Active Learning by Querying Informative and Representative Examples. IEEE Trans Pattern Anal Mach Intell 2014;36(10):1936–49.
[2] R. Yan, J. Yang, and A. Hauptmann, "Automatically labeling video data using multi-class active learning," Proceedings Ninth IEEE international conference on computer vision, pp. 516-523, 2003.
[3] H. T. Nguyen and A. Smeulders, "Active Learning Using Pre-clustering," Proceedings of the twenty-first international conference on Machine learning, pp. 79, 2004.
[4] K. Brinker, "Incorporating diversity in active learning with support vector machines," Proceedings of the 20th international conference on machine learning (ICML-03), pp. 59-66, 2003.
[5] X. Xia, P. Protopapas, and F. Doshi-Velez, "Cost-Sensitive Batch Mode Active Learning: Designing Astronomical Observation by Optimizing Telescope Time and Telescope Choice," Proceedings of the 2016 SIAM International Conference on Data Mining. Society for Industrial and Applied Mathematics, pp. 477-485, 2016.
[6] Chattopadhyay R, Wang Z, Fan W, Davidson I, Panchanathan S, Ye D. Batch Mode Active Sampling Based on Marginal Probability Distribution Matching. KDD : Proceedings International Conference on Knowledge Discovery & Data Mining 2013;7(3):1–25.
[7] S. Tong and E. Y. Chang, "Support vector machine active learning for image retrieval," Proceedings of the ninth ACM international conference on Multimedia, pp. 107-118, 2001.
[8] Freund Y, Seung HS, Shamir E, Tishby N. Selective sampling using the query by committee algorithm. Mach Learn 1997;28(2–3):133–68.
[9] H. S. Seung, M. Opper, and H. Sompolinsky, "Query by committee," Proceedings of the fifth annual workshop on Computational learning theory, 1992.
[10] Shi LX, Zhao YH, Tang J. Batch Mode Active Learning for Networked Data. ACM Trans Intell Syst Technol 2012;3(2):pp.
[11] S. C. H. Hoi, R. Jin, J. Zhu, and M. R. Lyu, "Batch mode active learning and its application to medical image classification," Proceedings of the 23rd international conference on Machine learning, pp.417–424, 2006.
[12] A. J. Joshi, F. Porikli, and N. Papanikolopoulos, "Multi-class batch-mode active learning for image classification," IEEE International Conference on Robotics & Automation, pp. 1873-1878, 2010.
[13] G. Schohn and D. Cohn, "Less is More: Active Learning with Support Vector Machines," Seventeenth International Conference on Machine Learning, pp. 839–846, 2000.
[14] Guo Y, Schuurmans D. Discriminative batch mode active learning. Adv Neural Inf Proces Syst 2007.

[15] X. Zhu, J. Lafferty, and Z. Ghahramani, "Combining active learning and semi-supervised learning using gaussian fields and harmonic functions," ICML 2003 workshop on the continuum from labeled to unlabeled data in machine learning and data mining, vol. 3, pp. 912–919, 2003.
[16] G. Dasarathy, R. Nowak, and X. J. C. S. Zhu, "S2: An Efficient Graph Based Active Learning Algorithm with Application to Nonparametric Classification," Conference on Learning Theory, pp.503-522, 2015.
[17] Baram Y, El-Yaniv R, Luz K. Online Choice of Active Learning Algorithms. J Mach Learn Res 2004;5(1):255–91.
[18] A. Bondu, V. Lemaire, and M. Boullé, "Exploration vs. exploitation in active learning : A Bayesian approach," In The 2010 International Joint Conference on Neural Networks (IJCNN), pp.1-7, 2010.
[19] T. T. Osugi, K. Deng, and S. D. Scott, "Balancing Exploration and Exploitation: A New Algorithm for Active Machine Learning," Fifth IEEE International Conference on Data Mining (ICDM'05), pp. 330–337, 2005.
[20] Cebron N, Berthold MRJDM, Discovery K. Active learning for object classification: from exploration to exploitation. Data Min Knowl Disc 2009;18(2):283–99.
[21] Haines TS, Xiang T. Active Rare Class Discovery and Classification Using Dirichlet Processes. Int J Comput Vis 2014;106(3):315–31.
[22] Loy CC, Xiang T, Gong S. Stream-based active unusual event detection. In: Asian Conference on Computer Vision. Springer; 2010. p. 161–75.
[23] C. C. Loy, T. M. Hospedales, T. Xiang, and S. Gong, "Stream-based joint exploration-exploitation active learning," In 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1560-1567, 2012.
[24] J. W. Stokes, J. Platt, J. Kravis, and M. Shilman, "Aladin: Active learning of anomalies to detect intrusions," In Proceeding of Neural Information Process System Conference 2007 Workshop on Machine Learning in Adversarial Environments for Computer Security MIT Press, pp. 12-13 2008.
[25] Tong S, Koller DJMLR. Support Vector Machine Active Learning with Applications to Text Classification. J Mach Learn Res 2002;2(1):999–1006.
[26] K. Simonyan and A. J. C. S. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," arXiv:1409.1556, 2014.
[27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770-778, 2016.
[28] R. J. C. S. Girshick, "Fast R-CNN," In Proceedings of the IEEE international conference on computer vision, pp. 1440-1448, 2015.
[29] Ren S, Girshick R, Girshick R, Sun A, Intelligence M. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. Adv Neural Inf Proces Syst 2017;39(6):1137–49.
[30] Z. Zhu, C. Yin, B. Qian, Y. Cheng, J. Wei, and F. Wang, "Measuring Patient Similarities via a Deep Architecture with Medical Concept Embedding," In IEEE International Conference on Data Mining, pp. 749-758, 2017.
[31] Y. J. E. A. Kim, "Convolutional Neural Networks for Sentence Classification," arXiv: 1408.5882. 2014.
[32] Yuan X, Zhang J, Yang L. IntSIM: An Integrated Simulator of Next-Generation Sequencing Data. IEEE Trans Biomed Eng 2016;64(2):441–51.
[33] C. Yin et al., "Deep similarity-based batch mode active learning with exploration-exploitation," 2017 IEEE International Conference on Data Mining (ICDM), pp. 575-584, 2017.
[34] Wang J, Wang L. A cooperative memetic algorithm with feedback for the energy-aware distributed flow-shops with flexible assembly scheduling. Comput Ind Eng 2022;168:108126.
[35] Pan Z, Lei D, Wang L. A knowledge-based two-population optimization algorithm for distributed energy-efficient parallel machines scheduling. IEEE Trans Cybern 2020;52(6):5051–63.
[36] Zhao F, Zhang H, Wang L. A pareto-based discrete jaya algorithm for multiobjective carbon-efficient distributed blocking flow shop scheduling problem. IEEE Trans Ind Inf 2022;19(8):8588–99.
[37] Zhao F, Di S, Wang L. A hyperheuristic with Q-learning for the multiobjective energy-efficient distributed blocking flow shop scheduling problem. IEEE Trans Cybern 2022;53(5):3337–50.
[38] Zhao F, Jiang T, Wang L. A reinforcement learning driven cooperative meta-heuristic algorithm for energy-efficient distributed no-wait flow-shop scheduling with sequence-dependent setup time. IEEE Trans Ind Inf 2022;19(7):8427–40.
[39] Zhao F, Xu Z, Wang L, Zhu N, Xu T. A population-based iterated greedy algorithm for distributed assembly no-wait flow-shop scheduling problem. IEEE Trans Ind Inf 2022;19(5):6693–705.