To Actively Initialize Active Learning

Yang, Yazhou ; Loog, Marco

DOI
10.1016/j.patcog.2022.108836
Publication date
2022
Document Version
Final published version
Published in
Pattern Recognition

Pattern Recognition 131 (2022) 108836

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

To Actively Initialize Active Learning

Yazhou Yang[a,*], Marco Loog[b,c]

[a] College of Advanced Interdisciplinary Studies, National University of Defense Technology, China
[b] Pattern Recognition Laboratory, Delft University of Technology, Van Mourik Broekmanweg 6, Delft 2628 XE, The Netherlands
[c] DIKU, University of Copenhagen, Universitetsparken 5, DK-2100, Denmark

ARTICLE INFO

Article history:
Received 2 November 2021
Revised 2 June 2022
Accepted 3 June 2022
Available online 12 June 2022

Keywords:
active learning
active initialization
nearest neighbor criterion
minimum nearest neighbor distance

ABSTRACT

Though much effort has been spent on designing new active learning algorithms, little attention has been paid to the initialization problem of active learning, i.e., how to find a set of labeled samples which contains at least one instance per category. This work identifies the initialization of active learning as a separate and novel research problem, reviews existing methods that can be adapted to be used for this task and, in addition, proposes a new active initialization criterion: the Nearest Neighbor Criterion. Experiments on 16 benchmark datasets verify that the novel method often finds an initialization set with fewer queried samples than other methods do.

1. Introduction

These days, we are witnessing a sharp increase in the amount of training data used in classification and regression tasks. Though the availability of large input data tends to boost the performance of machine learning models, it also leads to a big challenge: manually labeling samples can be very time-consuming and expensive [1]. Active learning has been proposed to tackle this challenge by querying only the most informative subsets from the whole data and maintaining good learning performance.

Most active learning approaches, which we refer to as supervised, need an initial labeled subset to start the active learning cycle based on true labels obtained from a human annotator. The main reason for the need of labels is that the selection criteria often depend on a trained classifier. For example, the simple margin method [2] trains a classifier on the initial labeled data and then chooses the sample nearest to its decision boundary. There are also unsupervised active learning methods, which do not use label information for sample selection. Transductive experimental design (TED) [3] and graph-based variance minimization methods [4,5], for instance, minimize the expected variance of a statistical model for which one does not need any labels. Unsupervised methods cannot exploit any information coming from the labeling provided and their performance can typically be improved by utilizing label information. For example, Zhen and Yeung [6] introduced a supervised version of TED, which adds a regularization term to incorporate label information and Gu et al. [7] proposed a weighted TED where the weight is estimated by using the class probability. Both report performance better than the original TED, indicating that supervised versions are to be preferred over unsupervised ones.

A crucial issue for supervised active learning, however, is that these methods assume to have a labeled dataset to start with. Though considerable efforts have been spent on seeking new active learning strategies, little attention has been paid to the initialization of these supervised methods. As also suggested by [8], we can view active learning as a two-step process: 1) find an initial labeled subset using some initialization strategy and 2) subsequently employ any preferred regular active learning algorithm. This paper considers exactly the first initialization step and investigates how to find an initial labeled subset to start the active learning process.

The obvious way to initialize active learning is to randomly select unlabeled instances until a subset containing at least one instance of each class has been obtained. This is also the most common strategy found in the literature [2,9–18]. Oftentimes, it is simply assumed that there exists an initial set consisting of a fixed number of instances per category randomly chosen from the unlabeled samples [12,13,15,19]. This may, in many novel application areas, not be realistic, as we typically do not have labeled samples to start with [8]. We therefore consider a more reasonable setting where we start the initialization without any labeled samples and actively choose the initial instances to label. We still assume here that the number of classes is known beforehand and the initialization phase is stopped when the initial set contains at least one instance per class.

* Corresponding author.
E-mail addresses: yangyazhou@nudt.edu.cn (Y. Yang), m.loog@tudelft.nl (M. Loog).

https://doi.org/10.1016/j.patcog.2022.108836
0031-3203/

1.1. Contribution and Outline

Following a review of current methods that are, with some minor changes, applicable to the initialization problem, we introduce our own, novel approach. Note that, as the initialization problem as such has not really been studied before, all methods presented are new in a sense. Comparisons are carried out based on two different types of experiments on 16 benchmark datasets. More specifically, the next section starts off presenting some further background and related work. Section 3 then reviews various known approaches and techniques that can be readily adapted to initialize active learning. That same section introduces and discusses our new strategy, which we call Nearest Neighbor Criterion (NNC for short). Section 4 describes and analyses the first experiments in which the various initialization strategies are compared purely based on the initialization phase. Section 5 then investigates the impact of different initialization strategies on the whole active learning process. Section 6 concludes our work.

2. Related Work

This work focuses on pool-based active learning where a large pool of unlabeled samples are readily available for querying [1]. This setting is the most widely studied. Reference [1] provides an early overview of active learning, covering many of its ins and outs. More recent complementary surveys can be found in [20–22].

There exist two earlier studies concerning the Initialization of Active Learning (IAL for short): [23,24], which both rely on k-means clustering. The work in [24] is problematic as it needs to fix an initialization size (e.g. 10% of the data) a priori, which is unwanted as it cannot guarantee that all classes have been identified using such fixed relative size. [23] has a similar problem, as it just takes a single sample from every cluster. In the setting we consider, the initialization stage is terminated once at least one instance from each class has been selected, which [24] and [23] cannot guarantee.

As experimental design approaches are unsupervised active learning techniques, they can be directly used for IAL. Examples are the D-optimality based transductive experimental design (TED) [3], V-optimality based graph variance minimization (V-opt) [4], and the recently proposed Σ-optimality based graph variance minimization (Σ-opt) [5]. These methods select the representative samples which reduce the variance of a specific statistical model. The differences among them are the chosen optimality criterion, which measures the overall variance, and the actual model they rely on. Subsection 3.1 provides further specifics regarding the experimental design and clustering-based approaches.

Further work can be found in the area of rare category detection [25–31]. The aim in these settings is to use active learning to identify interesting and useful anomalies, which are assumed to be very rare and typically can be found in tiny classes. The selection procedure is terminated when at least one representative sample from each rare class has been found. For example, He and Carbonell [26] first performed density estimation and then selected the instance that leads to an expected maximum change in local density once queried. Haines and Xiang [27] proposed a criterion called pWrong, which selects samples most likely to be wrong (i.e., belonging to an unseen category). Hospedales et al. [29] introduced Gen/Disc, which adaptively switches generative and discriminative classifiers in the learning progress to jointly discover new categories and maintain good learning performance. Hospedales et al. [28] proposed a criterion called Dirichlet Process Expected Accuracy (DPEA for short) to unify active learning and active class discovery. Huang et al. [30] attempted to find more instances of rare categories by assuming that one labeled example of each rare category is already available. Constructing such an initial labeled set for rare categories remains unsolved, which implies that investigating ALI in this work is meaningful.

Note that IAL differs from rare category detection in two respects. First, rare category detection focuses on finding (useful and useless) anomalies from normal data points, while IAL does not make any assumption on whether or not anomalies exist. IAL only concentrates on seeking representative samples to start active learning. Second, rare category detection always assumes that the datasets are extremely imbalanced with large majority classes and relatively small rare classes. IAL does not make such imbalanced assumption.

There are also some alternatives to tackle the initialization issue by using additional datasets or pre-trained models [32–34]. For instance, we can integrate transfer learning with active learning by directly using the model trained on the different but related dataset for selecting uncertain samples from our dataset. This process can be terminated when at least one example per class has been founded. However, this kind of approaches normally requires additional labeled datasets or pre-trained models on large-scale datasets, which restricts its application for IAL.

3. Adapted Techniques and a New Strategy

We briefly discuss various approaches from Section 2 that can be used to initialize active learning, possibly after some minor adaptation. Subsequently, we present a new initialization criterion: Nearest Neighbor Criterion (NNC). First, however, we make the IAL setup considered more precise.

A totally unlabeled dataset $\mathbf{P} = \{x_i\}_{i=1}^n$ is available, where $x_i \in \mathbb{R}^d$ is a feature vector. $\mathbf{I}$ denotes the initialization set and $C(\mathbf{I})$ its number of classes. $\mathbf{U}$ is the remaining unlabeled data: $\mathbf{U} = \mathbf{P} \backslash \mathbf{I}$. Algorithm 1 presents a basic active initialization process in which

Algorithm 1 Actively Initialize Active Learning

Require: unlabeled data $\mathbf{U}$, number of classes c, initial set $\mathbf{I} = \emptyset$.
1: **while** $C(\mathbf{I}) < c$ **do**
2: Choose the sample x^* according to some initialization criterion.
3: Query it for its label y and update$\mathbf{I} = \mathbf{I} \cup \{x^*, y\}$, $\mathbf{U} \leftarrow \mathbf{U} \backslash \{x^*\}$;
4: **end while**
5: Start some preferred supervised active learning algorithm with initial labeled $\mathbf{I}$.

the total number of all classes is c. We sequentially select an unlabeled sample and ask for its labels from a human annotator. When all classes have at least one instance, the initialization phase is stopped, and one can switch to one's preferred supervised active learning algorithm.

3.1. Adapted Techniques

In this subsection, we present how to adapt unsupervised active learning approaches for the initialization task. k-means++ and TED are chosen as the adapted techniques due to their simplicity and efficiency. Moreover, they represent the clustering-based methods and optimal design-based approaches, respectively.

Clustering-based Approaches k-means is considered for its simplicity [24,35]. Specifically, we use k-means++ which smartly chooses the initial seeds for k-means and performs well in practice [36]. There is no easy way to set the parameter k, but we suggest the following procedure. We perform k-means++ with $k = c$ in which c is the number of classes and query one instance nearest to the centroid from each cluster. If there are still some categories undiscovered, we continue the aforementioned procedure on the remaining unlabeled data until all categories own at least

Algorithm 2 Initialization with k-means++

```
I = ∅;
while C(I) < c do
    Perform k-means++ with k = c onU and set i = 1;
    while C(I) < c&i ≤ k do
        Choose the sample x* closest to the centroid from ith cluster;
        Query its true label y and updateI = I ∪ {x*, y}, U ← U\{x*}, i = i + 1;
    end while
end while
```

one data points. Algorithm 2 shows the overall initialization process. The main difference between Algorithm 2 and the approaches from [24,35] is that the latter set k to equal a pre-defined budget and terminate the active annotation process when the budget is finished.

Optimal Design-based Approaches We show by example how to adapt, in a simple way, methods for experimental design for the initialization of active learning. The example criterion we use is TED [3]. In brief, TED minimizes the variance of a regularized least square model by solving the following optimization problem:

$$\min_{\mathbf{X}\subset\mathbf{P}} \quad \mathrm{Tr}(\mathbf{P}(\mathbf{X}^T\mathbf{X}+\lambda\mathbb{I})^{-1}\mathbf{P}^T), \tag{1}$$

where $\mathbf{X}$ represents a set of instances to be queried, λ is the regularization parameter, and $\mathbb{I}$ is the identity matrix. A sequential greedy solution to TED is to select instance x such that $\mathrm{Tr}(\mathbf{P}(\mathbf{X_{t+1}}^T\mathbf{X_{t+1}}+\lambda\mathbb{I})^{-1}\mathbf{P}^T)$ achieves its minimum value, where $\mathbf{X}_{t+1}=\mathbf{X}_t\cup x$ and t represents the t-th round selection and $\mathrm{Tr}(\cdot)$ is the trace. We query the instances one by one according to the TED criterion and terminate the initialization process when $C(\mathbf{I})$ is equal to c. Clearly, other criteria used for optimal experimental design can be employed in the same way.

3.2. Minimum Nearest Neighbor Distance

As a last method, we present one that is new, simple, and fast. We refer to it as the Nearest Neighbor Criterion (NNC). It sequentially queries the most representative instance from unlabeled data such that the overall distance between queried and unlabeled data achieves its minimum value. The main motivation behind this is that we want to minimize the dissimilarity between labeled data and unlabeled data such that these labeled data can well-represent the remaining unlabeled data. We use the total nearest neighbor distance as a measure of the dissimilarity between queried data and unlabeled data, which is defined as follows:

$$NND(\mathbf{I},\mathbf{U})=\sum_{u\in\mathbf{U}}\min_{x\in\mathbf{I}}\|u-x\|, \tag{2}$$

where $\|u-x\|$ denotes the Euclidean distance between an unlabeled instance u and a labeled instance x.

$NND(\mathbf{I},\mathbf{U})$ computes the sum of the Euclidean distance between each unlabeled data point $u\in\mathbf{U}$ and its corresponding nearest neighbor x chosen from queried data $\mathbf{I}$. $NND(\mathbf{I},\mathbf{U})$ obtains a relatively small value when each unlabeled sample is close to its nearest neighbor. In other words, if each unlabeled data point is similar to its nearest neighbor, e.g. $\min_{x\in\mathbf{I}}\|u-x\|$ is small, then the overall neighbor distance between $\mathbf{U}$ and $\mathbf{I}$ is small too. This also implies that $NND(\mathbf{I},\mathbf{U})$ can be considered as a measure of how well the queried data can represent the unlabeled data. The smaller the value of $NND(\mathbf{I},\mathbf{U})$, the more representative $\mathbf{I}$ is. Note that $NND(\mathbf{I},\mathbf{U})$ can be easily extended by using other kinds of distance measures. For instance, the Euclidean distance can be replaced by Manhattan distance, Hamming distance, Minkowski distance, and Mahalanobis distance. Furthermore, $NND(\mathbf{I},\mathbf{U})$ can be redefined by using some kind of set-to-set distance [37].

Therefore, to initialize active learning, we select an unlabeled sample s that leads to a minimum value of the dissimilarity between queried data and unlabeled data once labeled. In other words, we select an unlabel sample, denoted by s, as the next queried data point such that the nearest neighbor distance between queried data $\mathbf{I}\cup\{s\}$ and the remaining unlabeled data $\mathbf{U}\backslash\{s\}$ obtains its minimum value. Our nearest neighbor criterion (NNC) is defined as follows:

$$x^*=\arg\min_{s\in\mathbf{U}} NND(\mathbf{I}\cup\{s\},\mathbf{U}\backslash\{s\}). \tag{3}$$

Algorithm 3 provides the pseudo-code of NNC.

Algorithm 3 Initialization with NNC

```
I = ∅;
while C(I) < c do
    for each unlabeled sample x_i ∈ U do
        Assume that x_i is chosen as the next queried sample;
        Compute the nearest neighbor distance NND(I ∪ x_i, U\x_i);
    end for
    Choose the sample x* with minimum nearest neighbor distance using Eq.~??.
    Query its true label y and updateI = I ∪ {x*, y}, U ← U\{x*};
end while
```

NNC can be seen as a sequential clustering algorithm. The clustering problem we consider is defined as follows:

$$\arg\min_{\mathbf{S}\subset\mathbf{P}}\sum_{u\in\mathbf{P}\backslash\mathbf{S}}\min_{x\in\mathbf{S}}\|u-x\| \tag{4}$$

where $\mathbf{S}$ is a set of samples we want to query. Each instance in $\mathbf{S}$ can be seen as an independent cluster seed. These seeds are used to cluster the remaining unlabeled data points based on the pairwise Euclidean distance. The number of seeds increase by 1 each time a new instance is chosen. The proposed NNC indeed provides a sequential greedy optimization approach to the problem in Equation 4. In the first iteration, since the initial set $\mathbf{I}$ is empty, NNC chooses the first point which is closest to the mean of all data $\mathbf{P}$. NNC then selects the subsequent sample by minimizing the nearest neighbor distance using Equation 3.

NNC has some links with the earlier mentioned k-means++ algorithm [36]. k-means++ first selects a random data point as the first cluster seed, and then selects the subsequent cluster seed with probability proportional to their squared distance from the closest existing cluster seeds. Two aspects distinguish NNC from k-means++. First, NNC queries the sample nearest to the mean of the data as the first point while k-means++ randomly chooses the first seed. Secondly, NNC selects the subsequent point by minimizing the nearest neighbor distance whereas k-means++ randomly chooses the next point based on some pre-defined probability.

4. Initialization Phase Experiments

We test the performance of the proposed NNC method and the main other methods covered in the foregoing. We first describe the experimental setup and, afterwards, present the way we compare the approaches. This section only looks at the initialization phase. Section 5 studies how the initialization impacts subsequent supervised active learning.

4.1. Experimental Setup

NNC is compared with the following algorithms: random sampling, k-means++ [36] (See Algorithm 2), TED [3], Σ-opt [5],

Table 1
Datasets information after pre-processing: number of instances (n), feature dimensionality (d), number of class (c).

Dataset	n	d	c	Dataset	n	d	c
MNIST	1000	60	10	USPS	1000	60	10
CIFAR10	1000	57	10	GTSRB	1000	40	20
UCFsports	140	100	10	Isolet	1040	40	26
pendigits	1000	16	10	satimage	1000	36	6
segment	1000	19	7	vowel	990	10	11
dermatology	366	34	6	led_display	1000	7	10
yeast	1484	8	10	ecoli	336	7	8
lowres	531	50	9	semeion	1593	50	10

pWrong [27], Gen/Disc [29], DPEA [28]. In addition, we also compare with a variant of uncertainty sampling, which first uses NNC to find instances which belong to two different classes (in order to train a classifier) and then use maximum entropy [1,38] to adaptively choose the most uncertain sample. This method, what we refer to as MaxE, is considered because we want to see whether an easy adaptation of a classical supervised active learning methods (i.e., uncertainty sampling in this case) can be suitable for the initialization task.

Among the methods compared, pWrong, Gen/Disc, and DPEA are three state-of-the-art rare category detection algorithms. We do not show results of the V-optimality based approach [4] since we observed that, in general, it performs worse than Σ-opt [5] in our experiments.

Having little or even no labeled data basically makes it impossible to tune parameters for classifiers or active learning algorithms. For this reason our experiments are set up with prefixed parameters, wherever possible we take their default values. More specifically, for pWrong, Gen/Disc, and DPEA, and TED, we use the default parameters mentioned either in the original references or in codes provided by the authors. For the remaining algorithms which do not have pre-specified default parameters, i.e. Σ-opt and MaxE, we carefully set their parameters to the values that work well in our preliminary experiments. For Σ-opt, we use regularized Laplacian graph kernel built by 5-nearest neighbor algorithm where heat kernel weighting is used. For MaxE, we use L_2 regularized logistic regression with regularization parameter $C = 100$ as classifier. Since also k-means++ is affected by the first randomly chosen seed, we try 100 different random seeds and choose the one which has lowest within-cluster sum of point-to-centroid distances.

Sixteen multi-class datasets are used in our experiments. Eleven of them, e.g. pendigits, semeion, Isolet, vowel, are taken from the UCI Machine Learning Repository [39]. Pendigits and semeion are two handwritten digits recognition datasets. The former are pen-based handwritten digits from 250 samples, using a pressure sensitive tablet. The latter are scans of 1593 handwritten digits, where each digit was scaled to 16×16 pixels. Isolet and vowel are speech recognition data sets. The Isolet dataset consists of the 26 letters of the English alphabet, while the vowel dataset is composed of 11 vowels in British English. The yeast and ecoli datasets are used to predict the cellular localization of particular proteins. We also test on five vision datasets. MNIST [40] and USPS [41] are handwritten digit datasets. CIFAR10 [42] and GTSRB [43] are image classification datasets. UCFsports [44] consists of 10 different categories of human actions collected from various sports videos, where the pre-extracted Action Bank features [45] are used in our experiment. For the MNIST and the USPS dataset, we use the gray-scale pixel values as the features. The HOG feature are extracted [46] for CIFAR10 and the GTSRB dataset. For computational efficiency, we use random sub-sampling and principal component analysis (PCA) on some datasets to reduce the sample size and feature dimensionality. Table 1 presents the properties of each dataset after pre-processing.

For each dataset, we randomly choose half of the data as the unlabeled data with which the active initialization is conducted. The remaining data is used to evaluate the classification accuracy of the chosen initial set. We repeat the random partition 100 times and average the results. We use L_2 regularized logistic regression from the LIBLINEAR package [47] with regularization parameter $C = 100$ as classifier.

4.2. Results and Analysis

Table 2 shows the performance of each method in terms of the means and standard deviations of the number of queried samples required to initialize active learning. The smaller the value, the better the performance. All the experiments are repeated 100 times and we use a paired t-test at a 95% significance level to check for statistically significant differences. The method which obtains the best performance or performs comparably to the best model is highlighted in bold face and coloured. We also report the average score over all test sets and the average ranking of each algorithm (i.e. the Mean and Average Ranking in Table 2). Wins counts the number of datasets on which an algorithm behaves the best or acts comparably to the best and win/tie/loss reports the win/tie/loss counts of other methods versus random initialization based on the paired t-test as well. Average Accuracy reports the average classification accuracy of all methods over all the test sets.

First of all, we find that NNC obtains the best performance on most datasets except on the dermatology and ecoli dataset. It also outperforms other models in terms of Mean, Average Ranking and Wins. The average number of samples required for NNC is around 30 while the second best needs, about, 37 data points. NNC achieves a higher average ranking of 1.62, whereas the second best model DPEA gets a score of 4.38. NNC also shows a clear advantage over random initialization with a win/tie/loss count of 15/1/0, which means that it dose not perform worse than random sampling on the 16 test sets.

Secondly, among the remaining compared approaches, DPEA and Σ-opt also perform well on most datasets, achieving best performance on 8 and 7 datasets, respectively. They require about 37 or 38 initial data points on average. Though pWrong obtains a reasonable performance in terms of Mean, i.e., 37.41, it only performs among the best on a single dataset: MNIST. MaxE also performs well compared to random sampling, i.e., obtaining a win/tie/loss count of 9/6/1. TED and Gen/Disc obtain a slightly worse performance than NNC, DPEA, and Σ-opt. k-means++ demonstrates a very poor performance with respects to the average number of required samples. It needs around 54 instances on average to initialize active learning. This number far exceeds that of all other compared methods except random sampling. The main reason could be that it is difficult to set an appropriate k beforehand.

We observe that random sampling is surpassed by all compared approaches in terms of Mean and Average Ranking. The average number of required samples for random sampling is around 63, while this value for the second worst model, k-means++, is around 54. Note, however, that on the CIFAR10 dataset, random sampling is among the best performing. We also find that some methods, e.g. DPEA, Gen/Disc, k-means++, fail to outperform random initialization on some particular datasets. Overall, however, a random initialization seems ill-advised.

We also evaluate the relative improvement of the active initialization criteria over random sampling. We consider the ratio $\frac{n_{AI}}{n_R}$ where n_{AI} and n_R are the number of necessary queries of active initialization models and random sampling, respectively. Table 3 reports the medians and median absolute deviations of these ratios over 100 trials. We use a Wilcoxon signed-rank test at a 95% significance level to check whether there is a statistical significant difference between two models. The method which obtains the best

Table 2
Means and standard deviations of number of queried samples (details in the main text).

	Random	k-means+	TED	Σ-opt	pWrong	Gen/Disc	DPEA	MaxE	NNC
USPS	32.34 (±14.11)	52.44 (±35.03)	19.57 (±9.13)	**16.05 (±4.86)**	24.50 (±10.87)	56.57 (±31.43)	20.15 (±10.03)	30.52 (±11.58)	**15.38 (±3.78)**
MNIST	31.88 (±11.57)	47.60 (±22.44)	**21.15 (±6.82)**	23.60 (±6.71)	**21.06 (±5.72)**	47.12 (±26.56)	**21.37 (±10.34)**	23.34 (±7.65)	**22.24 (±4.91)**
CIFAR10	**29.36 (±11.31)**	32.05 (±11.00)	**28.61 (±11.59)**	**29.65 (±8.45)**	35.84 (±10.66)	50.84 (±23.64)	47.60 (±17.36)	**28.92 (±9.81)**	**28.85 (±11.10)**
UCFsports	31.48 (±11.13)	23.72 (±10.31)	17.86 (±5.31)	32.67 (±12.64)	32.86 (±12.77)	32.43 (±12.82)	32.86 (±12.77)	38.56 (±18.57)	**15.21 (±4.21)**
GTSRB	69.67 (±20.98)	70.15 (±25.32)	**57.88 (±15.61)**	**61.25 (±21.69)**	66.96 (±18.92)	75.38 (±22.17)	**62.53 (±18.45)**	**61.78 (±16.21)**	**60.25 (±21.22)**
Isolet	92.67 (±25.24)	102.81 (±37.88)	81.69 (±28.22)	81.30 (±24.84)	102.67 (±34.38)	90.54 (±26.90)	**74.32 (±22.33)**	93.70 (±29.71)	**71.57 (±23.65)**
pendigits	30.10 (±11.09)	47.67 (±27.89)	24.12 (±7.84)	30.02 (±12.37)	29.55 (±11.59)	20.80 (±5.98)	41.08 (±11.87)	28.36 (±15.23)	**18.45 (±4.89)**
satimage	20.91 (±10.19)	103.24 (±26.13)	23.15 (±13.19)	**9.34 (±1.97)**	19.56 (±9.46)	15.09 (±6.59)	**9.65 (±4.23)**	17.78 (±9.61)	**9.03 (±1.65)**
yeast	252.70 (±145.10)	87.20 (±44.23)	**66.09 (±40.30)**	**66.74 (±33.00)**	69.88 (±37.98)	71.94 (±46.74)	**58.95 (±27.32)**	106.44 (±116.38)	**62.08 (±33.95)**
segment	16.94 (±5.98)	20.95 (±8.03)	14.77 (±4.56)	**10.82 (±2.84)**	15.04 (±4.93)	26.79 (±10.07)	**10.93 (±3.42)**	18.26 (±7.84)	**10.47 (±2.72)**
vowel	34.65 (±13.34)	53.57 (±23.34)	79.84 (±60.82)	30.50 (±9.30)	34.36 (±9.39)	57.38 (±41.73)	28.60 (±10.70)	30.88 (±10.95)	**25.95 (±5.98)**
lowres	176.19 (±51.76)	108.75 (±82.48)	82.55 (±41.04)	**24.55 (±4.52)**	29.03 (±11.35)	73.48 (±18.22)	68.35 (±6.97)	78.14 (±31.87)	38.54 (±13.61)
dermatology	23.53 (±15.76)	12.10 (±7.03)	8.50 (±3.02)	25.45 (±10.44)	17.11 (±8.27)	11.09 (±4.54)	22.17 (±8.53)	10.96 (±4.74)	**6.46 (±1.35)**
led_display	29.63 (±12.91)	**20.47 (±8.25)**	67.20 (±30.40)	23.09 (±7.74)	24.51 (±7.79)	27.35 (±10.54)	**19.19 (±8.48)**	27.37 (±11.10)	**18.95 (±8.43)**
ecoli	107.65 (±38.01)	49.86 (±46.25)	87.89 (±40.23)	74.18 (±46.68)	47.16 (±28.47)	62.86 (±44.15)	**38.89 (±29.90)**	**32.68 (±21.49)**	45.93 (±35.13)
semeion	29.00 (±12.58)	23.58 (±12.58)	26.61 (±7.88)	75.73 (±19.84)	28.52 (±9.18)	23.69 (±8.44)	37.64 (±12.47)	20.58 (±6.11)	**18.57 (±5.13)**
Mean (± std)	63.04 (±66.01)	53.51 (±32.21)	44.22 (±29.01)	38.43 (±24.49)	37.41 (±23.60)	46.46 (±24.54)	37.14 (±20.37)	40.52 (±28.67)	29.25 (±20.43)
Average Accuracy	62.91 (±17.61)	64.09 (±17.13)	62.46 (±17.16)	62.51 (±16.02)	61.66 (±16.51)	61.77 (±17.33)	64.39 (±17.57)	61.38 (±17.38)	64.08 (±17.13)
Average Ranking	6.94	6.69	4.44	4.50	5.19	6.19	4.38	5.06	1.62
Wins	1	1	4	7	1	0	8	3	14
win/tie/loss	-	7/2/7	11/3/2	11/4/1	8/6/2	7/4/5	11/2/3	9/6/1	15/1/0

Table 3
Medians and deviations of ratio of number of queries relative to random sampling (details in the main text).

	Random	k-means+	TED	Σ-opt	pWrong	Gen/Disc	DPEA	MaxE	NNC
USPS	1.00 (±0.00)	1.49 (±0.81)	0.63 (±0.26)	**0.52 (±0.20)**	0.72 (±0.27)	1.64 (±0.80)	0.59 (±0.20)	0.88 (±0.28)	**0.52 (±0.16)**
MNIST	1.00 (±0.00)	1.35 (±0.52)	**0.68 (±0.21)**	0.75 (±0.20)	**0.64 (±0.18)**	1.41 (±0.45)	**0.67 (±0.17)**	0.74 (±0.24)	**0.71 (±0.21)**
CIFAR10	**1.00 (±0.00)**	**1.14 (±0.36)**	**0.96 (±0.33)**	**1.09 (±0.34)**	1.24 (±0.32)	1.63 (±0.57)	1.69 (±0.49)	**0.94 (±0.29)**	**0.95 (±0.26)**
UCFsports	1.00 (±0.00)	0.71 (±0.26)	0.58 (±0.15)	1.00 (±0.39)	1.03 (±0.40)	1.06 (±0.38)	1.03 (±0.40)	1.13 (±0.44)	**0.48 (±0.14)**
GTSRB	**1.00 (±0.00)**	1.02 (±0.30)	**0.83 (±0.24)**	**0.81 (±0.24)**	**0.94 (±0.24)**	1.18 (±0.33)	**0.86 (±0.21)**	**0.91 (±0.26)**	**0.89 (±0.28)**
Isolet	1.00 (±0.00)	1.08 (±0.36)	0.87 (±0.30)	0.85 (±0.19)	1.07 (±0.31)	0.93 (±0.26)	**0.75 (±0.21)**	1.02 (±0.31)	**0.79 (±0.23)**
pendigits	1.00 (±0.00)	1.46 (±0.68)	0.83 (±0.27)	1.02 (±0.35)	1.03 (±0.34)	0.69 (±0.17)	1.53 (±0.48)	0.87 (±0.30)	**0.65 (±0.16)**
satimage	1.00 (±0.00)	5.13 (±1.91)	1.07 (±0.45)	**0.50 (±0.17)**	0.93 (±0.37)	0.68 (±0.23)	**0.50 (±0.17)**	0.93 (±0.37)	**0.47 (±0.17)**
yeast	1.00 (±0.00)	0.39 (±0.18)	**0.27 (±0.13)**	**0.28 (±0.16)**	**0.31 (±0.17)**	**0.27 (±0.15)**	**0.25 (±0.11)**	0.37 (±0.19)	**0.24 (±0.13)**
segment	1.00 (±0.00)	1.31 (±0.45)	0.87 (±0.27)	**0.67 (±0.21)**	0.87 (±0.30)	1.64 (±0.58)	**0.64 (±0.22)**	1.02 (±0.38)	**0.65 (±0.18)**
vowel	1.00 (±0.00)	1.58 (±0.61)	1.75 (±1.00)	0.95 (±0.30)	1.04 (±0.28)	1.42 (±0.65)	**0.84 (±0.30)**	0.84 (±0.26)	**0.80 (±0.24)**
lowres	1.00 (±0.00)	0.49 (±0.29)	0.41 (±0.14)	**0.13 (±0.03)**	0.16 (±0.04)	0.41 (±0.10)	0.38 (±0.08)	0.45 (±0.16)	0.22 (±0.08)
dermatology	1.00 (±0.00)	0.55 (±0.22)	0.43 (±0.19)	1.20 (±0.49)	0.78 (±0.42)	0.55 (±0.22)	1.09 (±0.50)	0.53 (±0.27)	**0.34 (±0.13)**
led_display	1.00 (±0.00)	**0.68 (±0.21)**	2.38 (±0.95)	0.77 (±0.27)	0.88 (±0.27)	0.98 (±0.37)	**0.63 (±0.28)**	0.93 (±0.29)	**0.60 (±0.24)**
ecoli	1.00 (±0.00)	0.29 (±0.15)	0.79 (±0.30)	0.66 (±0.34)	0.40 (±0.19)	0.52 (±0.32)	**0.32 (±0.20)**	**0.25 (±0.13)**	0.39 (±0.26)
semeion	1.00 (±0.00)	0.74 (±0.26)	0.91 (±0.27)	2.90 (±0.84)	1.03 (±0.31)	0.85 (±0.24)	1.33 (±0.47)	0.74 (±0.23)	**0.66 (±0.20)**
Mean (± std)	1.00 (±0.00)	1.21 (±1.12)	0.89 (±0.52)	0.88 (±0.61)	0.82 (±0.30)	0.99 (±0.46)	0.82 (±0.42)	0.78 (±0.25)	0.59 (±0.22)
Average Ranking	6.62	6.44	4.69	4.44	5.44	6.38	4.19	4.88	1.94
Wins	2	2	4	7	3	1	9	3	14
win/tie/loss	-	7/1/8	9/4/3	9/5/2	5/9/2	7/3/6	10/2/4	6/8/2	15/1/0

performance or performs comparably to the best model is highlighted in bold face and coloured. NNC clearly improves upon random sampling and other models in terms of mean, wins, and average ranking. It outperforms random sampling on 15 datasets except the CIFAR10 on which a tie is reached. The mean of the medians of the ratios of NNC to random sampling is 0.59, which means that NNC can save 41% annotation cost. DPEA, MaxE, and pWrong also demonstrate good performances, reducing the cost by about a fifth. *k*-means++ shows performance poorer than random sampling, obtaining an average ratio of 1.21 and a win/tie/loss count of 7/1/8.

Figure 1 shows the plots of the average number of discovered classes with respects to the number of queried samples. The x-axis is the number of queried samples and the y-axis is the exponentiation with number two as the base and average number of classes as the exponent. We use the exponential function for a better visualization of the behaviour of different methods in the latter part of the initialization stage. Obviously, the faster the increase, the better the method. Note that the maximal number of queried instances (on the horizontal axis) differs from dataset to dataset. We observe that the NNC clearly outperforms other approaches on most datasets, i.e., the USPS, UCFsports, Isolet, pendigits, satimage, segment, vowel, dermatology, and semeion. Random sampling performs among the best on the CIFAR10 dataset, and obtains poor performance on the remaining datasets. Σ-opt behaves the best on the lowres dataset while becoming the worst one on the semeion dataset. Gen/Disc performs the worst on several datasets, i.e. the USPS, CIFAR10, GTSRB, and segment. TED is the slowest one to discover classes on the vowel and led_display dataset. Another observation to make is that NNC is never worst performing over the 16 test sets.

For a better understanding of the characteristic of different initialization criteria, we generate a synthetic dataset and provide the preference map [20]. As shown in Figure, this synthetic dataset consists of points from six clusters with several outliers in the upper left corner and points with the same color share the same class labels. This preference map is generated by creating the synthetic 2D dataset 100 times with different random seeds and recording the locations of the first six sample points chosen by different initialization methods. We choose to count the first six queried points since this synthetic dataset is composed of six clusters and any initialization criteria requires at least six samples to finish the initialization task. We calculate the local density of chosen points and show the density map with pseudo-colors. Blue and red colors indicate the lowest and highest density, respectively. Means and standard deviations of number of queried samples on the synthetic dataset is also reported in Table 4. We observe that *k*-means++, pWrong, DEPA, and NNC perform among the best.

As shown in Figure, *k*-means++ and NNC perform similarly and tend to select the points near the centre of the clusters. This coincides with the incremental clustering characteristic of NNC and demonstrates that NNC prefers the representative samples. The phenomenon that pWrong prefers points in the upper left corner implies that pWrong may be easily affected by the outliers. NNC and *k*-means++ are less likely to be influenced by the outliers. TED with a default linear kernel fails on this synthetic dataset and is less likely to query the points near the centre. We find that this situation can be solved by using a radial basis function kernel TED. MaxE under-performs on this dataset and ignores the points in the bottom left corner. The reason may be that MaxE queries the first several pints from the central points and then gets stuck in the poorly estimated uncertainty. Σ-opt and Gen/Disc performs similarly to each other and the central points are selected regardless of their distribution density.

When an initial set has been constructed, i.e., at least one instance has been queried from each class, we evaluate the classification accuracy on the test set to evaluate how informative the queried initial set is. Figure 3 illustrates the average classification accuracy of the initial set chosen by different initialization criteria w.r.t. the average number of queried samples over 100 trials. The point in the upper left corner means that an algorithm has the overall best performance since it achieves the highest classification accuracy with the smallest number of queried samples. On the contrary, the point in the lower right corner indicates that this methods performs poorly even with a large number of initial instances.

We find that NNC obtains relatively high accuracy with a reasonable number of queried instances on most datasets. In Figure, NNC achieves the second highest accuracy with around 15 samples while *k*-means++ has the best accuracy with about 50 instances. NNC has a similar accuracy to *k*-means++ on the MNIST and segment dataset, but it only needs about half of the samples. NNC also performs well in terms of the average accuracy on CIFAR10, pendigits, segment, dermatology, led_display, and semeion.

Table 2 also reports the average classification accuracy of all methods over all test sets. NNC obtains about 64% accuracy with less than 30 samples. DPEA has a similar performance to NNC

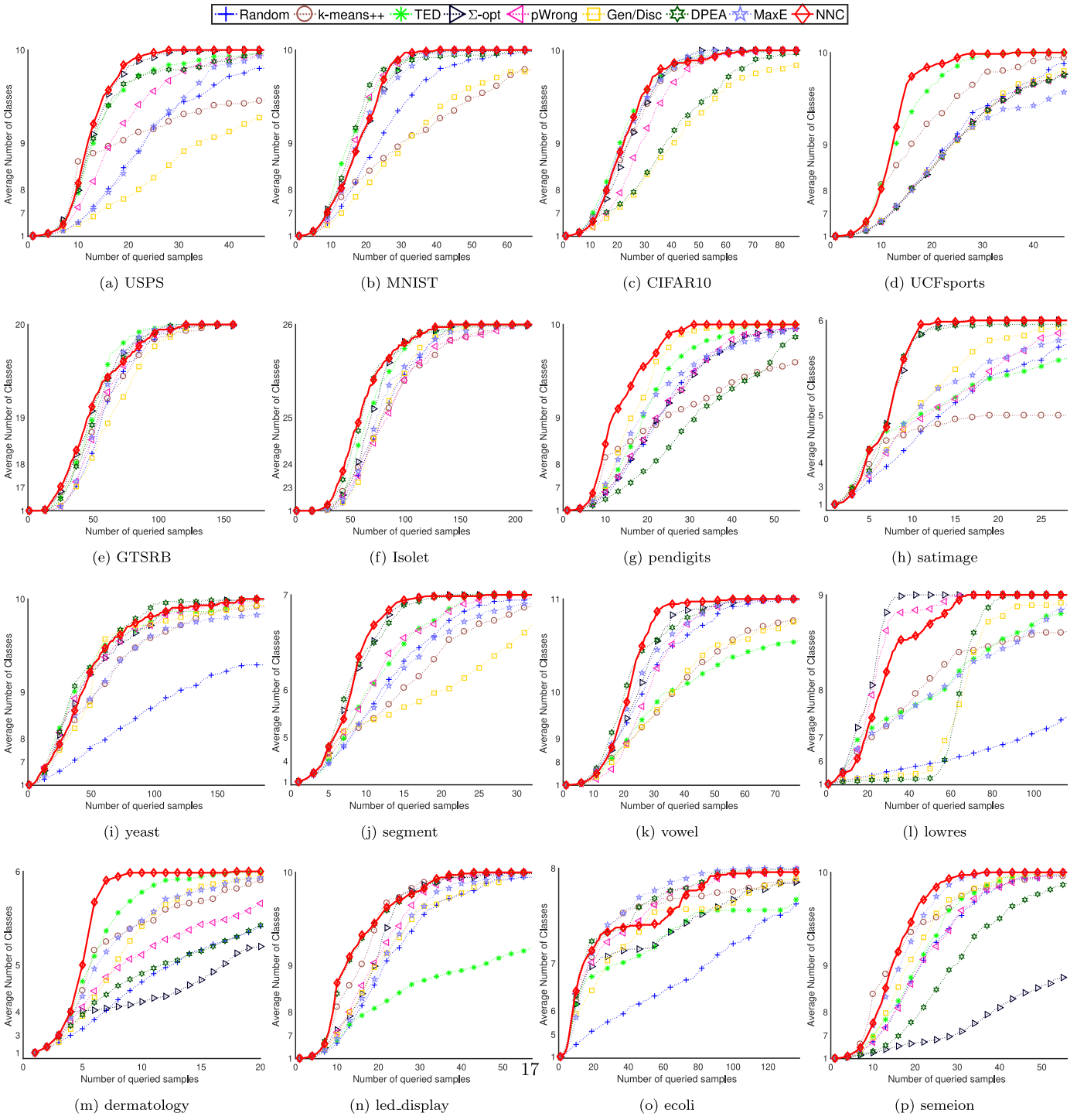


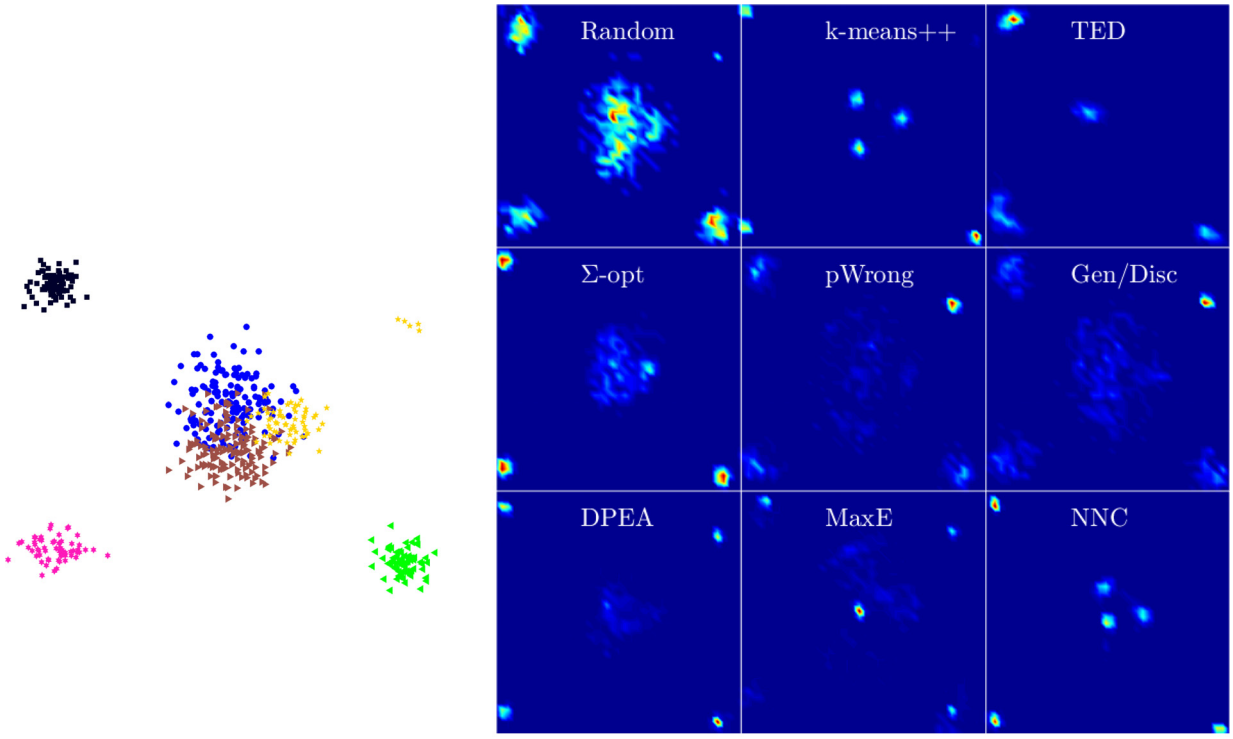
Fig. 1. Plots of the exponential of the average number of discovered classes with number two as the base w.r.t. the number of queried samples. On the x-axis is the number of queried samples and on the y-axis is the exponential of the average number of classes with base two.

in terms of the average accuracy, but it requires around 37 instances. *k*-means++ obtains the second best accuracy 64.09% but with a cost of requiring 24 more samples than NNC. Overall, the new method shows good performance in terms of the number of queried samples and also the classification accuracy. DPEA is the second best initialization strategy, while Σ-opt slightly performs worse than DPEA.

Table 5 reports the average computational cost of finding an initial labeled subset on 16 datasets. On each dataset we repeat the experiments 20 times and report the average time of constructing

Table 4
Means and standard deviations of number of queried samples on the synthetic dataset.

Random	k-means+	TED	Σ-opt	pWrong	Gen/Disc	DPEA	MaxE	NNC
17.88 (±7.65)	**7.08 (±2.70)**	119.80 (±14.16)	7.88 (±2.63)	**6.82 (±1.54)**	9.87 (±4.63)	**7.13 (±1.33)**	154.32 (±111.43)	**6.92 (±1.03)**



(a) a synthetic dataset (b) Preference maps

Fig. 2. Preference maps of a synthetic dataset. (a) shows the distribution of a synthetic dataset and (b) demonstrates the preference maps of the compared initialization criteria on dataset (a). Points with different colors belong to different categories.

Table 5
Computational cost comparison of constructing an initial labeled set (in seconds).

	Random	k-means+	TED	Σ-opt	pWrong	Gen/Disc	DPEA	MaxE	NNC
USPS	0.01	2.33	0.04	201.76	1.76	1.98	46.63	0.10	0.29
MNIST	0.01	2.52	0.04	295.25	1.65	1.74	45.96	0.07	0.48
CIFAR10	0.01	1.91	0.07	390.75	1.82	1.72	106.00	0.08	0.76
UCFsports	0.00	0.52	0.00	0.42	0.67	0.75	23.42	0.14	0.02
GTSRB	0.01	2.76	0.16	801.78	2.55	3.56	348.55	0.40	2.06
Isolet	0.01	3.22	0.21	1017.85	2.36	5.77	301.78	0.60	1.65
pendigits	0.02	1.76	0.08	400.43	1.80	0.62	34.59	0.07	0.35
satimage	0.02	7.71	0.08	127.56	2.64	1.06	12.70	0.07	0.17
yeast	0.01	5.27	0.24	738.28	1.93	1.69	41.12	0.51	3.82
segment	0.01	0.93	0.03	41.36	1.21	0.34	5.51	0.03	0.13
vowel	0.01	2.31	0.18	245.76	1.61	1.41	22.49	0.06	0.48
lowres	0.01	3.96	0.06	64.49	1.18	3.30	115.98	0.39	0.32
dermatology	0.00	0.38	0.01	7.44	0.48	0.30	10.07	0.01	0.02
led_display	0.00	0.64	0.13	181.36	0.84	0.50	9.18	0.04	0.33
ecoli	0.00	1.12	0.02	6.58	0.43	0.57	7.74	0.02	0.09
semeion	0.01	3.00	0.12	3359.18	1.60	0.53	69.91	0.08	0.71
Mean	0.01	2.52	0.09	492.52	1.53	1.61	75.10	0.17	0.73

an initial set. The experiments are constructed with MATLAB 9.1 on an Intel(R) Core 3.0GHz i9-9980XE CPU PC with 128 GB memory. We see that random sampling, TED, and MaxE are highly efficient due to their simplicity. NNC is less cost-effective than random sampling, TED, and MaxE but clearly outperform other compared methods. We also note that Σ-opt and DPEA have a remarkably heavy computational cost. The reason is that Σ-opt has to compute the inverse of matrix multiple times and DPEA requires to retrain the classifiers w.r.t. all possible classes and all unlabeled samples.

5. Initialization Impact on Active Learning

When the initialization stage of active learning is completed, we can continue with the second step, which is employing any regular active learning algorithm until the labeling budget has been spent or some stopping criterion is satisfied. One may wonder, however, how the initial data set affects the active learning strategy, as the quality of initial labeled data has an impact on the acquisitions of the subsequently queried samples.

We briefly investigate this issue empirically. One of the challenges is that the number of possible experiments easily gets out of hand, as results may vary depending on the labeling budget allowed and the subsequent active learning strategy employed. We include our investigation for completeness, but keep it necessarily limited. The specific setup is as follows.

Uncertainty sampling and a variance maximization method (MVAL) [48] are chosen as the subsequent active learning criteria after initialization for two reasons: (1) uncertainty sampling

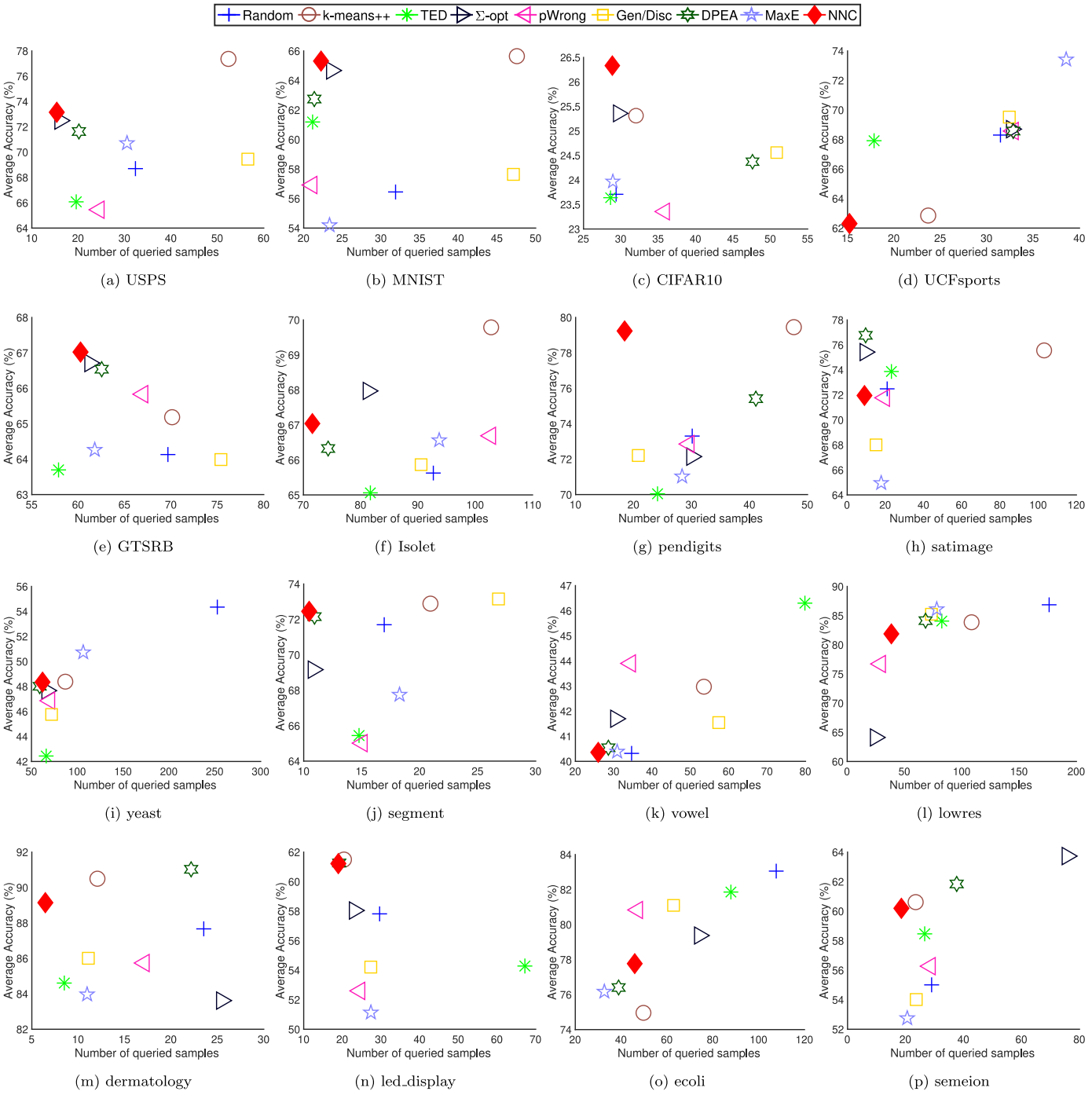


Fig. 3. Illustration of the average accuracy w.r.t. the number of queried samples. On the x-axis is the average number of queried samples and on the y-axis is the average accuracy over 100 trials.

is one of the simplest and most widely used active learning algorithms [1] and is among the top performers in an earlier extensive benchmark experiment that the authors conducted [20]; (2) MVAL demonstrates a superior performance over its competitors including uncertainty sampling with a drawback of high computational cost. For uncertainty sampling, we employ logistic regression as the base classifier and use entropy as the uncertainty measure [1,38]. For each dataset, we start with the different initialization strategies and switch to the uncertainty sampling on the condition that all classes have at least one labeled sample and the labeling budget is not finished. It is therefore possible that underperforming criteria exhaust all the budget on the initialization, leaving none for subsequent uncertainty sampling. The actual choice of budget may differs slightly from dataset to dataset. For most datasets, we choose 100 because accuracies do not increase much after annotating 100 instances. For GTSRB, CIFAR10, and vowel, we stop earlier because of the heavy computation cost they incur. Like in the foregoing experiment, L_2 regularized logistic regression with a regularization parameter $C = 100$ from the LIBLINEAR package [47] is used. Following previous works [48–51], we use the area under the learning curve (ALC) as our performance measure. We repeat the experiments 100 times and report the average results in Table 6 and 7.

Table 6
Performance in terms of the areas under the learning curve (ALC; details in the main text). Uncertainty sampling is chosen as the subsequent active learning method.

	Random	k-means+	TED	Σ-opt	pWrong	Gen/Disc	DPEA	MaxE	NNC
USPS	0.738	0.767	0.777	0.797	0.756	0.678	0.795	0.742	**0.801**
MNIST	0.610	0.640	0.668	0.670	0.650	0.590	**0.673**	0.627	**0.673**
CIFAR10	0.254	0.259	0.256	0.261	0.243	0.237	0.238	0.251	**0.265**
UCFsports	0.627	0.665	**0.710**	0.633	0.631	0.629	0.631	0.642	0.701
GTSRB	0.535	0.582	0.593	**0.618**	0.582	0.555	0.591	0.558	0.612
Isolet	0.458	0.523	0.494	0.504	0.489	0.465	0.525	0.469	**0.532**
pendigits	0.693	0.732	0.690	0.694	0.694	0.710	0.696	0.702	**0.752**
satimage	0.709	0.686	**0.757**	**0.748**	0.719	0.693	**0.755**	0.657	0.727
yeast	0.454	0.463	0.433	0.468	0.452	0.444	0.469	0.473	**0.481**
segment	0.729	0.702	0.743	0.775	0.743	0.688	0.785	0.712	**0.792**
vowel	0.346	0.324	0.354	0.353	**0.366**	0.309	0.357	0.351	0.352
lowres	0.784	0.795	0.777	0.731	0.781	0.777	0.802	0.800	**0.810**
dermatology	0.854	0.901	0.912	0.841	0.885	0.885	0.868	0.906	**0.922**
led_display	0.572	**0.616**	0.502	0.587	0.575	0.566	0.607	0.563	0.609
ecoli	0.794	0.797	0.804	0.809	**0.814**	0.811	**0.816**	**0.813**	0.812
semeion	0.532	0.579	0.562	0.496	0.536	0.540	0.562	0.542	**0.585**
Average ALC	0.606	0.627	0.627	0.624	0.62	0.599	0.636	0.613	0.652
Average Ranking	7.31	4.75	4.75	4.69	5.25	7.38	3.56	5.5	1.81
Wins	0	1	2	2	2	0	3	1	10
win/tie/loss	–	13/0/3	12/1/3	12/2/2	13/1/2	6/1/9	15/0/1	12/0/4	16/0/0

Table 7
Performance in terms of the areas under the learning curve (ALC; details in the main text). MVAL is chosen as the subsequent active learning method.

	Random	k-means+	TED	Σ-opt	pWrong	Gen/Disc	DPEA	MaxE	NNC
USPS	0.732	0.765	0.777	0.796	0.754	0.676	0.787	0.744	**0.798**
MNIST	0.617	0.642	0.671	0.677	0.661	0.593	0.676	0.646	**0.682**
CIFAR10	0.246	0.261	0.249	0.260	0.240	0.238	0.244	0.250	**0.265**
UCFsports	0.574	0.670	**0.706**	0.631	0.631	0.638	0.631	0.643	0.700
GTSRB	0.515	0.587	0.597	**0.622**	0.586	0.544	0.594	0.563	0.615
Isolet	0.443	0.523	0.495	0.507	0.492	0.462	0.529	0.471	**0.535**
pendigits	0.701	0.739	0.712	0.710	0.711	0.730	0.705	0.712	**0.763**
satimage	0.713	0.686	**0.763**	0.755	0.734	0.716	**0.764**	0.677	0.733
yeast	0.450	0.466	0.431	0.469	0.450	0.441	0.473	0.473	**0.480**
segment	0.729	0.705	0.750	0.783	0.741	0.700	0.788	0.717	**0.795**
vowel	0.341	0.327	0.357	0.363	**0.370**	0.332	0.366	0.361	0.364
lowres	0.780	0.795	0.778	0.741	0.790	0.781	0.804	0.801	**0.813**
dermatology	0.849	0.904	0.914	0.843	0.889	0.897	0.874	0.910	**0.925**
led_display	0.564	**0.612**	0.508	0.593	0.581	0.575	**0.608**	0.569	**0.611**
ecoli	0.778	0.795	0.803	0.807	**0.813**	**0.810**	**0.812**	**0.810**	**0.810**
semeion	0.546	0.603	0.576	0.499	0.561	0.564	0.573	0.564	**0.611**
Average ALC	0.599	0.63	0.631	0.628	0.625	0.606	0.639	0.619	0.656
Average Ranking	7.75	4.75	4.81	4.88	5.25	6.75	3.75	5.25	1.81
Wins	0	1	2	1	2	1	3	1	12
win/tie/loss	–	13/0/3	13/1/2	13/1/2	14/1/1	8/2/6	15/0/1	14/0/2	16/0/0

Figure 4 shows the average accuracy of the whole active learning procedure w.r.t. the number of queried samples over 100 trials with uncertainty sampling as the base active learning algorithm. The color coding of the lines indicated what fraction of queries moved into the second stage at that point. That is, the line is blue if most of the 100 trials are still in the initialization stage, the line is red of all trials have already moved on to the regular active learning strategy. Colors in between indicate fraction between 0 (blue) and 1 (red).

From Table 2 and 6, we find that uncertainty sampling initialized with NNC achieves the best overall performance in terms of the average ALC and average ranking. In addition, it performs among the best over 10 datasets while the second best performer DPEA only wins over 3 datasets. Furthermore, it outperforms the active learner initialized with random initialization over all the 16 test datasets. However, other initialization criteria may lead to performance worse than random initialization on multiple datasets. For instance, Gen/Disc's performance is surpassed by random initialization on USPS and MNIST. Similarly, on semeion, Σ-opt performs clearly worse than random initialization. Uncertainty sampling initialized with NNC surpasses the one initialized with MaxE on most datasets except on ecoli where the latter performs slightly better. Note that MaxE uses uncertainty sampling to find other initial labeled instances after the first two classes are found by employing NNC. The clearly better performance of NNC over MaxE could indicate that first employing some well-performing initialization strategies and then switching to regular active learning algorithms is more likely to succeed than directly exploiting some active learning algorithms to find all the required labeled instances.

We observe similar results in Table 7 with MVAL as the base active learning method. MVAL initialized with NNC performs among the best over 12 datasets and obtain excellent scores in terms of the average ALC and average ranking. A further observation from Table 6 and 7 is that uncertainty sampling and MVAL obtain similar performance when they are started with the same initialization criterion. For instances, when initialized with NNC, the average ALC of uncertainty sampling is 0.652 and that of MVAL is 0.656. Similar situations occur with other initialization criteria, e.g., uncertainty sampling and MVAL with DPEA obtain 0.636 and 0.639,

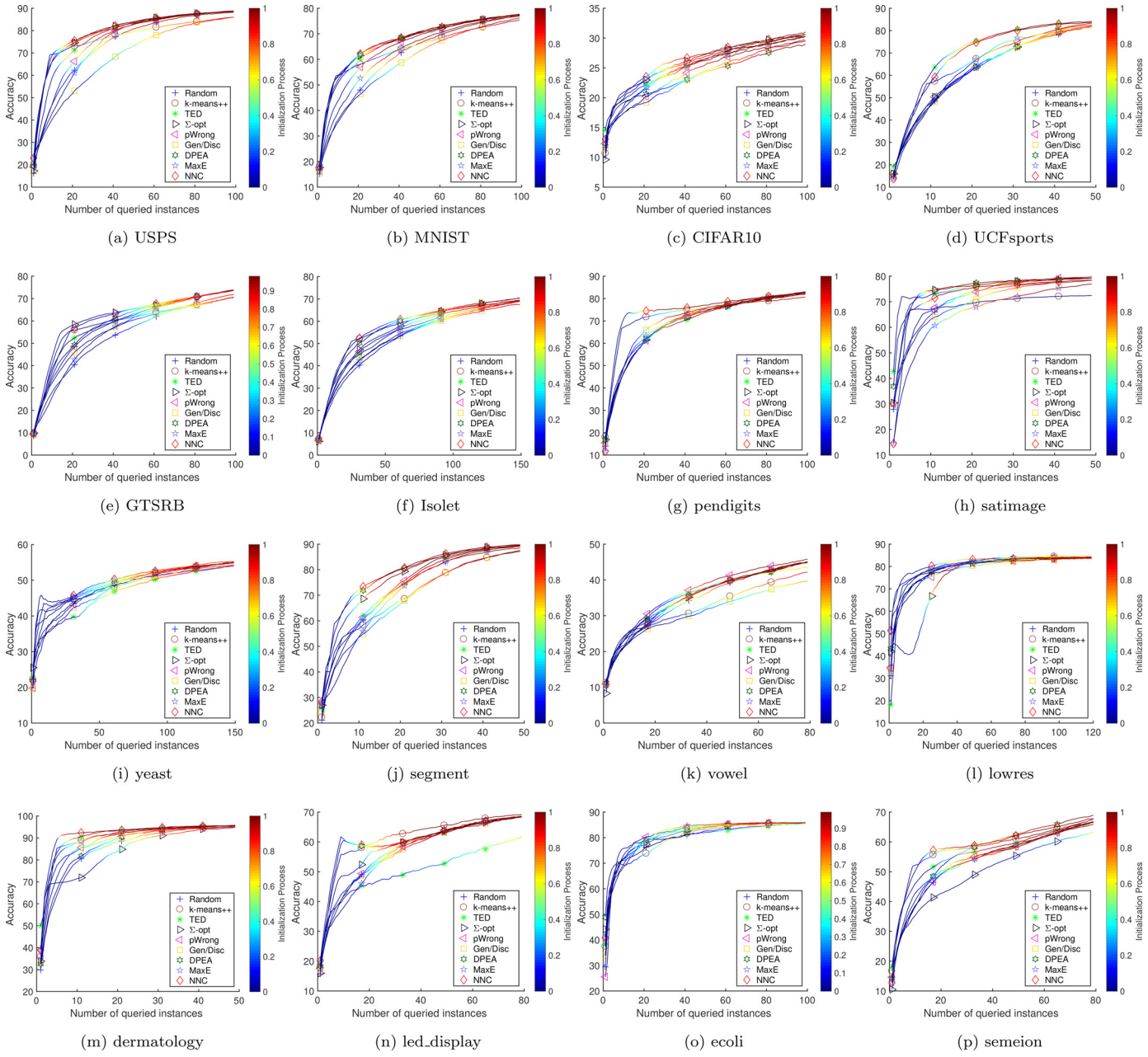


Fig. 4. Plots of the average accuracy w.r.t. the number of queried samples. On the x-axis is the average number of queried samples and on the y-axis is the average accuracy over 100 trials. Each line is colored based on its initialization process. Uncertainty sampling is used as the base active learning method.

respectively. This implies that the initialization strategies exert greater influence on the learning performance over the subsequent active learning algorithms.

To further demonstrate the effectiveness and robustness of the proposed method, we also report the comparison results by employing a linear SVM and a non-linear extreme learning machine. More specially, we use a linear SVM from the LIBSVM package [52] with regularization parameter C being set to 10 in all the experiments. For extreme learning machine, we use the code available online[1] with a radial basis function kernel of default parameter. We present the overall average performances of the compared methods using three different classifiers (i.e. logistic regression, linear SVM, and extreme learning machine) in combination with two active learning approaches (i.e. uncertainty sampling and MVAL) in Tables 8 and 9. The details of the performances on each single dataset with SVM and extreme learning machine as the classifiers are available in the appendix.

We observe that the proposed method NNC still achieves the best performances in terms of Average ALC, Average Ranking, and Wins, regardless of the classifier applied or the active learning method employed. For example, as shown in Table 8, in case of uncertainty sampling being utilized, NNC obtains the best scores of average ALC, average ranking, and wins, no matter which classifier is applied. The same phenomenon is observed in Table 9 in which MVAL is employed as the subsequent active learning method. Note that when extreme learning machine is applied, the average ALC of the compared methods performed slightly worse than that of

[1] http://www.extreme-learning-machines.org.

Table 8
Performance in terms of the areas under the learning curve (ALC) using different classifiers on 16 datasets. Uncertainty sampling is chosen as the subsequent active learning method.

Classifier		Random	k-means+	TED	Σ-opt	pWrong	Gen/Disc	DPEA	MaxE	NNC
Logistic Regression	Average ALC	0.606	0.627	0.627	0.624	0.62	0.599	0.636	0.613	0.652
	Average Ranking	7.31	4.75	4.75	4.69	5.25	7.38	3.56	5.5	1.81
	Wins	0	1	2	2	2	0	3	1	10
Linear SVM	Average ALC	0.62	0.643	0.64	0.636	0.632	0.612	0.648	0.628	0.668
	Average Ranking	6.94	4.38	5.06	4.75	5.25	7.06	4.12	5.81	1.62
	Wins	0	1	1	1	2	1	4	1	10
Extreme Learning Machine	Average ALC	0.584	0.606	0.601	0.602	0.593	0.573	0.616	0.588	0.632
	Average Ranking	7.12	4.19	4.38	4.75	5.81	7.31	3.5	6.06	1.88
	Wins	0	1	1	2	2	0	6	0	9

Table 9
Performance in terms of the areas under the learning curve (ALC) using different classifiers on 16 datasets. MVAL is chosen as the subsequent active learning method.

Classifier		Random	k-means+	TED	Σ-opt	pWrong	Gen/Disc	DPEA	MaxE	NNC
Logistic Regression	Average ALC	0.599	0.63	0.631	0.628	0.625	0.606	0.639	0.619	0.656
	Average Ranking	7.75	4.75	4.81	4.88	5.25	6.75	3.75	5.25	1.81
	Wins	0	1	2	1	2	1	3	1	12
Linear SVM	Average ALC	0.611	0.643	0.642	0.639	0.636	0.62	0.65	0.634	0.671
	Average Ranking	7.69	4.56	4.69	4.75	5.38	6.62	3.94	5.69	1.69
	Wins	0	1	2	2	1	0	3	1	11
Extreme Learning Machine	Average ALC	0.579	0.606	0.602	0.606	0.598	0.582	0.617	0.594	0.633
	Average Ranking	7.44	4.44	4.56	4.38	5.94	6.75	4	5.81	1.69
	Wins	0	1	1	3	2	0	4	0	11

using other classifiers. The possible reason could be that extreme learning machine is not fine-tuned in the active learning setting due to a lack of additional labeled validation set. However, we observe that NNC outperforms other methods, even in case of extreme learning machine being used as the classifier. For example, MVAL initialized with NNC still obtains the best results over 11 datasets with extreme learning machine being the classifier. Similarly, uncertainty sampling starting with NNC performs among the best on 9 datasets when extreme learning machine is employed. This indicates that the proposed method NNC is substantially robust to the classifier used and active learning method employed.

All in all, we find that the behaviour of different initialization strategies in the initialization task is quite related to their performances on the whole active learning process. For example, NNC and DPEA are the top two performers in the task of initialization (see Table 2) and are also the two best ones in terms of average ALC (see Table 6). Moreover, the average ranking of these methods in Table 2 and 6 are similar to each other. This suggests that a more efficient initialization criterion is more likely to promote the overall performance of an active learning algorithm. A poor performing initialization strategy has a high chance of decreasing the performance.

6. Discussion and Conclusion

We investigated how to find a labeled set to initialize active learning algorithms with as few annotations as possible, while at the same time, the initial set consists of at least one instance from each class. This is a relevant task that nevertheless has not really been studied before. Next to all the existing methods that could be readily adapted for the initialization task, a new criterion, NNC, was proposed. It selects the samples which minimizes the dissimilarity between unlabeled data and the queried data that has been labeled, where the dissimilarity is measured by the overall nearest neighbor distance. Experiments demonstrate that the number of queried samples obtained by this method is clearly less than that of the algorithms compared to. In addition, the initially data selected by NNC shows good performance with respect to the classification accuracy in comparison to other approaches. Moreover, we investigated the impact of different initialization strategies on the complete active learning process, concluding that starting active learning with a more efficient initialization criterion is more likely to obtain a better performance at lower cost. We can all in all conclude that NNC should be considered the current method of preference for IAL.

There are several other questions that may warrant further investigations. It is absolutely of interest to consider what to do if one does know the number of classes in advance. All the initialization criteria presented in this work make this assumption, giving us a clear criterion of when to stop the initialization stage. In real-world applications, however, it is possible that we do not have prior knowledge about how many categories the data contains. In that case, we have to consider finding some, possibly more elaborate criteria that can also be used to decide on terminating the active initialization process.

To us, however, the primary challenge seems to construct methods for IAL that can all in all guarantee to lead to performances better than random sampling. Similar question concerning active learning have been raised before, e.g. in [53]. Active learning and its initialization stage is something one sets out to do once. If, however, an active learner cannot, at least to some extent, guarantee that it will not be worse than random sampling, actually using such strategies in realistic applications may never happen.

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.

Acknowledgement

This work was supported in part by National Natural Science Foundation of China under Grant No. 61906208.

Appendix. >Appendix

Table A.1
Performance in terms of the areas under the learning curve (ALC) with SVM as the classifier. Uncertainty sampling is chosen as the subsequent active learning method.

	Random	k-means+	TED	Σ-opt	pWrong	Gen/Disc	DPEA	MaxE	NNC
USPS	0.736	0.764	0.763	0.791	0.745	0.669	0.787	0.735	**0.794**
MNIST	0.627	0.656	0.687	0.690	0.666	0.604	**0.693**	0.645	**0.694**
CIFAR10	0.267	0.272	0.269	0.273	0.257	0.251	0.252	0.264	**0.278**
UCFsports	0.605	0.641	**0.689**	0.605	0.604	0.601	0.604	0.622	0.679
GTSRB	0.535	0.586	0.583	**0.619**	0.573	0.547	0.586	0.553	0.613
Isolet	0.493	0.560	0.532	0.545	0.515	0.505	0.556	0.508	**0.573**
pendigits	0.748	0.774	0.747	0.745	0.748	0.764	0.740	0.754	**0.808**
satimage	0.721	0.693	0.758	0.761	0.730	0.705	**0.770**	0.671	0.739
yeast	0.448	0.469	0.425	0.464	0.441	0.439	0.468	0.469	**0.479**
segment	0.732	0.722	0.735	0.772	0.738	0.691	**0.785**	0.707	0.781
vowel	0.420	0.397	0.408	0.430	**0.435**	0.363	0.432	0.425	**0.435**
lowres	0.792	0.802	0.801	0.775	0.803	0.802	0.813	**0.824**	**0.824**
dermatology	0.842	0.898	0.899	0.831	0.874	0.874	0.857	0.891	**0.914**
led_display	0.571	**0.618**	0.502	0.585	0.569	0.557	0.606	0.553	0.607
ecoli	0.779	0.779	0.795	0.795	**0.800**	**0.798**	**0.803**	0.794	0.798
semeion	0.607	0.656	0.642	0.490	0.614	0.619	0.613	0.630	**0.672**
Average ALC	0.62	0.643	0.64	0.636	0.632	0.612	0.648	0.628	0.668
Average Ranking	6.94	4.38	5.06	4.75	5.25	7.06	4.12	5.81	1.62
Wins	0	1	1	1	2	1	4	1	10
win/tie/loss	–	12/1/3	10/3/3	11/2/3	11/3/2	7/0/9	13/1/2	11/1/4	16/0/0

Table A.2
Performance in terms of the areas under the learning curve (ALC) with SVM as the classifier. MVAL is chosen as the subsequent active learning method.

	Random	k-means+	TED	Σ-opt	pWrong	Gen/Disc	DPEA	MaxE	NNC
USPS	0.730	0.762	0.766	**0.791**	0.747	0.668	0.779	0.738	**0.792**
MNIST	0.632	0.660	0.689	0.696	0.675	0.605	0.692	0.662	**0.703**
CIFAR10	0.260	0.273	0.263	0.273	0.253	0.252	0.261	0.264	**0.280**
UCFsports	0.545	0.646	**0.687**	0.606	0.605	0.614	0.605	0.623	0.679
GTSRB	0.511	0.589	0.585	**0.619**	0.575	0.538	0.589	0.557	0.613
Isolet	0.477	0.558	0.532	0.546	0.515	0.505	0.557	0.509	**0.574**
pendigits	0.750	0.776	0.765	0.759	0.761	0.782	0.749	0.763	**0.817**
satimage	0.724	0.693	**0.766**	0.764	0.744	0.732	**0.775**	0.688	0.738
yeast	0.448	0.470	0.421	0.464	0.440	0.437	0.471	0.469	**0.479**
segment	0.729	0.720	0.741	0.779	0.740	0.707	**0.787**	0.714	0.781
vowel	0.412	0.398	0.408	0.441	0.437	0.392	0.442	0.437	**0.451**
lowres	0.788	0.802	0.802	0.784	0.810	0.808	0.814	**0.824**	**0.826**
dermatology	0.839	0.900	0.900	0.834	0.879	0.889	0.863	0.896	**0.919**
led_display	0.562	**0.610**	0.508	0.589	0.571	0.569	0.606	0.559	**0.608**
ecoli	0.761	0.781	0.795	0.794	**0.802**	0.798	**0.800**	0.792	0.795
semeion	0.604	0.658	0.642	0.491	0.622	0.629	0.616	0.642	**0.673**
Average ALC	0.611	0.643	0.642	0.639	0.636	0.62	0.65	0.634	0.671
Average Ranking	7.69	4.56	4.69	4.75	5.38	6.62	3.94	5.69	1.69
Wins	0	1	2	2	1	0	3	1	11
win/tie/loss	–	13/0/3	13/1/2	13/2/1	14/0/2	10/0/6	14/2/0	13/1/2	16/0/0

Table A.3
Performance in terms of the areas under the learning curve (ALC) with extreme learning machine as the classifier. Uncertainty sampling is chosen as the subsequent active learning method.

	Random	k-means+	TED	Σ-opt	pWrong	Gen/Disc	DPEA	MaxE	NNC
USPS	0.717	0.753	0.766	0.780	0.744	0.659	**0.786**	0.730	0.783
MNIST	0.604	0.625	0.667	0.667	0.652	0.588	**0.677**	0.632	0.672
CIFAR10	0.273	0.275	0.277	0.277	0.259	0.251	0.252	0.270	**0.281**
UCFsports	0.237	0.248	**0.268**	0.241	0.241	0.237	0.241	0.248	0.261
GTSRB	0.508	0.569	0.554	**0.599**	0.549	0.519	0.563	0.520	0.593
Isolet	0.465	0.531	0.501	0.510	0.488	0.472	0.523	0.476	**0.541**
pendigits	0.760	0.783	0.763	0.758	0.759	0.779	0.744	0.771	**0.829**
satimage	0.719	0.699	0.753	0.768	0.729	0.703	**0.776**	0.670	0.750
yeast	0.449	0.479	0.413	0.466	0.420	0.423	0.470	0.460	**0.488**
segment	0.712	0.685	0.716	0.750	0.712	0.671	**0.764**	0.692	**0.762**
vowel	0.408	0.376	0.409	0.426	**0.439**	0.364	0.431	0.421	0.427
lowres	0.758	0.762	0.758	0.709	0.736	0.746	**0.787**	0.754	**0.787**
dermatology	0.820	0.880	0.876	0.815	0.845	0.846	0.837	0.860	**0.892**
led_display	0.555	**0.610**	0.472	0.568	0.531	0.520	0.595	0.517	0.594
ecoli	0.783	0.791	0.799	**0.805**	**0.799**	0.795	**0.806**	0.788	**0.804**
semeion	0.586	0.639	0.616	0.498	0.590	0.595	0.601	0.602	**0.648**
Average ALC	0.584	0.606	0.601	0.602	0.593	0.573	0.616	0.588	0.632
Average Ranking	7.12	4.19	4.38	4.75	5.81	7.31	3.5	6.06	1.88
Wins	0	1	1	2	2	0	6	0	9
win/tie/loss	–	13/0/3	11/3/2	12/1/3	10/2/4	6/1/9	14/0/2	11/1/4	16/0/0

Table A.4
Performance in terms of the areas under the learning curve (ALC) with extreme learning machine as the classifier. MVAL is chosen as the subsequent active learning method.

	Random	k-means+	TED	Σ-opt	pWrong	Gen/Disc	DPEA	MaxE	NNC
USPS	0.707	0.749	0.768	0.779	0.747	0.664	0.780	0.736	**0.785**
MNIST	0.603	0.622	0.656	0.662	0.653	0.591	0.665	0.641	**0.668**
CIFAR10	0.267	0.280	0.268	0.279	0.255	0.260	0.260	0.272	**0.282**
UCFsports	0.229	0.245	**0.265**	0.240	0.238	0.239	0.238	0.247	0.259
GTSRB	0.489	0.572	0.557	**0.601**	0.554	0.509	0.566	0.525	0.594
Isolet	0.452	0.530	0.502	0.511	0.490	0.471	0.526	0.477	**0.542**
pendigits	0.758	0.784	0.777	0.768	0.770	0.794	0.753	0.778	**0.831**
satimage	0.723	0.699	0.761	0.771	0.740	0.730	**0.780**	0.689	0.744
yeast	0.447	0.481	0.407	0.465	0.418	0.425	0.469	0.460	**0.485**
segment	0.711	0.687	0.728	0.761	0.724	0.691	**0.772**	0.702	0.767
vowel	0.403	0.376	0.410	**0.439**	**0.440**	0.388	0.439	0.431	**0.440**
lowres	0.754	0.761	0.759	0.725	0.745	0.751	**0.788**	0.756	**0.788**
dermatology	0.818	0.880	0.877	0.817	0.852	0.860	0.845	0.866	**0.895**
led_display	0.543	**0.593**	0.481	0.573	0.541	0.538	0.593	0.532	**0.595**
ecoli	0.770	0.792	0.801	**0.804**	**0.801**	0.795	**0.801**	0.785	0.798
semeion	0.584	0.640	0.615	0.499	0.597	0.603	0.603	0.610	**0.650**
Average ALC	0.579	0.606	0.602	0.606	0.598	0.582	0.617	0.594	0.633
Average Ranking	7.44	4.44	4.56	4.38	5.94	6.75	4	5.81	1.69
Wins	0	1	1	3	2	0	4	0	11
win/tie/loss	–	13/0/3	12/2/2	13/1/2	12/1/3	8/2/6	14/0/2	12/1/3	16/0/0

References

[1] B. Settles, Active learning literature survey, University of Wisconsin, Madison 52 (55-66) (2010) 11.
[2] S. Tong, D. Koller, Support vector machine active learning with applications to text classification, The Journal of Machine Learning Research 2 (2002) 45–66.
[3] K. Yu, J. Bi, V. Tresp, Active learning via transductive experimental design, in: Proceedings of the 23rd International Conference on Machine Learning, ACM, 2006, pp. 1081–1088.
[4] M. Ji, J. Han, A variance minimization criterion to active learning on graphs, in: Artificial Intelligence and Statistics, 2012, pp. 556–564.
[5] Y. Ma, R. Garnett, J. Schneider, σ-optimality for active learning on gaussian random fields, in: Advances in Neural Information Processing Systems, 2013, pp. 2751–2759.
[6] Y. Zhen, D.-Y. Yeung, Sed: supervised experimental design and its application to text classification, in: Proceedings of the 33rd ACM SIGIR, ACM, 2010, pp. 299–306.
[7] Q. Gu, T. Zhang, J. Han, Batch-mode active learning via error bound minimization, in: Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence, AUAI Press, 2014, pp. 300–309.
[8] D. Kottke, A. Calma, D. Huseljic, G. Krempl, B. Sick, Challenges of reliable, realistic and comparable active learning evaluation, in: Proceedings of the Workshop and Tutorial on Interactive Adaptive Learning, 2017, pp. 2–14.
[9] X. Zhu, J. Lafferty, Z. Ghahramani, Combining active learning and semi-supervised learning using gaussian fields and harmonic functions, in: ICML 2003 workshop on The Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining, 2003, pp. 58–65.
[10] N. Roy, A. McCallum, Toward optimal active learning through sampling estimation of error reduction, in: Proceedings of the 18th ICML, 2001, pp. 441–448.
[11] K. Brinker, Incorporating diversity in active learning with support vector machines, in: Proceedings of the 20th International Conference on Machine Learning, 2003, pp. 59–66.
[12] Y. Baram, R.E. Yaniv, K. Luz, Online choice of active learning algorithms, Journal of Machine Learning Research 5 (Mar) (2004) 255–291.
[13] X. You, R. Wang, D. Tao, Diverse expected gradient active learning for relative attributes, IEEE Transactions on Image Processing 23 (7) (2014) 3203–3217.
[14] A.I. Schein, L.H. Ungar, Active learning for logistic regression: an evaluation, Machine Learning 68 (3) (2007) 235–265.
[15] Q. Shi, B. Du, L. Zhang, Spatial coherence-based batch-mode active learning for remote sensing image classification, IEEE Transactions on Image Processing 24 (7) (2015) 2037–2050.
[16] D. Lowell, Z.C. Lipton, B.C. Wallace, Practical obstacles to deploying active learning, arXiv preprint arXiv:1807.04801 (2018).
[17] F. Zhdanov, Diverse mini-batch active learning, arXiv preprint arXiv:1901.05954 (2019).
[18] S. Belharbi, I. Ben Ayed, L. McCaffrey, E. Granger, Deep active learning for joint classification & segmentation with weak annotator, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2021, pp. 3338–3347.

[19] Y. Yang, Z. Ma, F. Nie, X. Chang, A.G. Hauptmann, Multi-class active learning by uncertainty sampling with diversity maximization, International Journal of Computer Vision 113 (2) (2015) 113–127.
[20] Y. Yang, M. Loog, A benchmark and comparison of active learning for logistic regression, Pattern Recognition 83 (2018) 401–415.
[21] H. Hino, Active learning: Problem settings and recent developments, arXiv preprint arXiv:2012.04225 (2020).
[22] X. Zhan, H. Liu, Q. Li, A.B. Chan, A comparative survey: Benchmarking for pool-based active learning, in: Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, 2021, pp. 4679–4686, doi:10.24963/ijcai.2021/634.
[23] J. Kang, K.R. Ryu, H.-C. Kwon, Using cluster-based sampling to select initial training set for active learning in text classification, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 2004, pp. 384–388.
[24] R. Hu, B. Mac Namee, S.J. Delany, Off to a good start: Using clustering to select the initial training set in active learning, 23rd FLAIRS, 2010.
[25] D. Pelleg, A.W. Moore, Active learning for anomaly and rare-category detection, in: Advances in Neural Information Processing Systems, 2005, pp. 1073–1080.
[26] J. He, J.G. Carbonell, Nearest-neighbor-based active learning for rare category detection, in: Advances in Neural Information Processing Systems, 2008, pp. 633–640.
[27] T.S.F. Haines, T. Xiang, Active learning using dirichlet processes for rare class discovery and classification, in: British Machine Vision Conference, 2011, pp. 1–11.
[28] T.M. Hospedales, S. Gong, T. Xiang, A unifying theory of active discovery and learning, in: European Conference on Computer Vision, 2012, pp. 453–466.
[29] T.M. Hospedales, S. Gong, T. Xiang, Finding rare classes: Active learning with generative and discriminative models, IEEE Transactions on Knowledge and Data Engineering 25 (2) (2013) 374–386.
[30] H. Huang, Q. Yan, W. Lu, H. Lin, Y. Gao, L. Chen, Leri: Local exploration for rare–category identification, IEEE Transactions on Knowledge and Data Engineering 32 (9) (2020) 1761–1772.
[31] Z. Liu, S. Hu, Y. Yin, J. Chen, K. Chiew, L. Zhang, Z. Wu, Interactive rare-category-of-interest mining from large datasets, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, 2020, pp. 4965–4972.
[32] L.E. Dor, A. Halfon, A. Gera, E. Shnarch, L. Dankin, L. Choshen, M. Danilevsky, R. Aharonov, Y. Katz, N. Slonim, Active learning for bert: an empirical study, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2020, pp. 7949–7962.
[33] H. Shao, Query by diverse committee in transfer active learning, Frontiers of Computer Science 13 (2) (2019) 280–291.
[34] M. Yuan, H.-T. Lin, J. Boyd-Graber, Cold-start active learning through self-supervised language modeling, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2020, pp. 7935–7948.
[35] V. Souza, R.G. Rossi, G.E. Batista, S.O. Rezende, Unsupervised active learning techniques for labeling training sets: An experimental evaluation on sequential data, Intelligent Data Analysis 21 (5) (2017) 1061–1095.
[36] D. Arthur, S. Vassilvitskii, k-means++: The advantages of careful seeding, in: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms, Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035.
[37] J. Zhao, J. Han, L. Shao, Unconstrained face recognition using a set-to-set distance measure on deep learned features, IEEE Transactions on Circuits and Systems for Video Technology 28 (10) (2017) 2679–2689.
[38] D.D. Lewis, W.A. Gale, A sequential algorithm for training text classifiers, in: Proceedings of the 17th ACM SIGIR, 1994, pp. 3–12.
[39] M. Lichman, UCI machine learning repository, 2013, (????). http://archive.ics.uci.edu/ml.
[40] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE 86 (11) (1998) 2278–2324.
[41] J.J. Hull, A database for handwritten text recognition research, IEEE Transactions on Pattern Analysis and Machine Intelligence 16 (5) (1994) 550–554.
[42] A. Krizhevsky, G. Hinton, Learning multiple layers of features from tiny images, Technical Report, University of Toronto 1 (4) (2009).
[43] J. Stallkamp, M. Schlipsing, J. Salmen, C. Igel, The german traffic sign recognition benchmark: a multi-class classification competition, in: In International Joint Conference on Neural Networks, IEEE, 2011, pp. 1453–1460.
[44] M.D. Rodriguez, J. Ahmed, M. Shah, Action mach a spatio-temporal maximum average correlation height filter for action recognition, in: Computer Vision and Pattern Recognition (CVPR), 2008 IEEE Conference on, IEEE, 2008, pp. 1–8.
[45] S. Sadanand, J.J. Corso, Action bank: A high-level representation of activity in video, in: IEEE Conference on CVPR, 2012, IEEE, 2012, pp. 1234–1241.
[46] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, volume 1, IEEE, 2005, pp. 886–893.
[47] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, C.-J. Lin, Liblinear: A library for large linear classification, The Journal of Machine Learning Research 9 (2008) 1871–1874.
[48] Y. Yang, M. Loog, A variance maximization criterion for active learning, Pattern Recognition 78 (2018) 358–370.
[49] D. Mazzoni, K. Wagstaff, Active learning in the presence of unlabelable examples, Technical Report, Jet Propulsion Laboratory, National Aeronautics and Space Administration, 2004.
[50] M. Culver, D. Kun, S. Scott, Active learning to maximize area under the roc curve, in: Sixth International Conference on Data Mining (ICDM'06), IEEE, 2006, pp. 149–158.
[51] D.J. Cook, N.C. Krishnan, Activity learning: discovering, recognizing, and predicting human behavior from sensor data, John Wiley & Sons, 2015.
[52] C.-C. Chang, C.-J. Lin, Libsvm: a library for support vector machines, ACM Transactions on Intelligent Systems and Technology 2 (3) (2011) 27:1–27:27.
[53] M. Loog, Y. Yang, An empirical investigation into the inconsistency of sequential active learning, in: 23rd International Conference on Pattern Recognition, IEEE, 2016, pp. 210–215.

Yazhou Yang received the B.S. degree in information system engineering, the M.S degree of control science and engineering from the National University of Defense Technology, Changsha, China, in 2011 and 2013, respectively. He got a Ph.D. degree from the Pattern Recognition Laboratory at Delft University of Technology, Delft, the Netherlands, in 2018. He is currently a research assistant at the National University of Defense Technology. His current research interests include active learning, semi-supervised learning, deep learning and image classification.

Marco Loog received an M.Sc. degree in mathematics from Utrecht University and a Ph.D. degree from the Image Sciences Institute. He worked as a scientist at the IT University of Copenhagen, the University of Copenhagen, and Nordic Bioscience. While still affiliated with the University of Copenhagen, he now primarily resides at Delft University of Technology to research and teach. His principal scholarly interest is with supervised learning in all sorts of shapes and sizes.