



Top K representative: a method to select representative samples based on K nearest neighbors

Kai Yang² · Yi Cai¹ · Zhiwei Cai³ · Haoran Xie⁴ · Tak-Lam Wong⁵ · Wai Hong Chan⁴

Received: 1 April 2017 / Accepted: 1 December 2017 / Published online: 12 December 2017
© Springer-Verlag GmbH Germany, part of Springer Nature 2017

Abstract

Short text categorization involves the use of a supervised learning process that requires a large amount of labeled data for training and therefore consumes considerable human labor. Active learning is a way to reduce the number of manually labeled samples in traditional supervised learning problems. In active learning, the number of samples is reduced by selecting the most representative samples to represent an entire training set. Uncertainty sampling is a means of active learning but is easily affected by outliers. In this paper, a new sampling method called Top K representative (TKR) is proposed to solve the problem caused by outliers. However, TKR optimization is a nondeterministic polynomial-time hardness (NP-hard) problem, making it challenging to obtain exact solutions. To tackle this problem, we propose a new approach based on the greedy algorithm, which can obtain approximate solutions, and thereby achieve high performance. Experiments show that our proposed sampling method outperforms the existing methods in terms of efficiency.

Keywords Active learning · Text categorization

1 Introduction

With the growing popularity of the Internet, shorter texts are being generated by users and services on such platforms as Twitter and Facebook [10]. Moreover, techniques of short text classification have a number of applications, such as sentiment classification [18] and semantic analysis [16]. Therefore, there is considerable demand for short text classification or categorization [1]. Short text classification requires a large number of labeled samples for training [2], because it involves a supervised learning process [26]. However, the volume of short texts generated by Web services

is enormous, and manually labeling all these data items is time consuming and expensive. Labeling work takes a long time, creating a bottleneck in most short text classification problems.

Active learning is a widely used technique to solve the above problem [36]. The main idea underlying active learning is to choose representative samples by using machine learning algorithms to generate a small set. We only need to label this small set with fewer samples as the training set. We can thus save a large amount of time and money. The main task of this paper is to apply active learning to reduce manual labeling work in supervised classification tasks.

A number of active learning algorithms have been developed, e.g., Uncertainty sampling [21] or some density-based selection methods [18, 39, 40]. Uncertainty sampling is a sampling method that is widely used in natural language processing tasks such as text categorization and grammar matching. However, its limitation is that some outliers (i.e., abnormal samples) may be selected as representative samples, which affects the quality of its results. In another word, outliers have great negative impact on the performance of uncertainty sampling. To solve this problem, sampling methods based on the density distribution of the training set have been proposed [18, 39, 40]. As outliers are generally located in low-density space, these methods involve the selection

The preliminary version of this article was published in ASC 2017 conjunction with BIGCOMP 2017 [38].

✉ Yi Cai
ycai@scut.edu.cn

- ¹ South China University of Technology, Guangzhou, China
- ² City University of Hong Kong, Hong Kong, Hong Kong
- ³ South China University of Technology, Guangzhou, China
- ⁴ The Education University of Hong Kong, Hong Kong, Hong Kong
- ⁵ Douglas College, New Westminster, Canada

of samples from high-density space to avoid choosing outliers. However, such methods also have some drawbacks. Samples located along the borders of categories are not necessarily located in high-density space. These methods therefore cannot accurately select samples along the borders of categories, while samples in the borders are important for classification.

In this paper, we propose a sampling algorithm called Top K Representative (TKR). This algorithm is inspired by the K-Nearest Neighbor (KNN) algorithm [3, 8], an important classification algorithm. In KNN, each sample can be represented by k samples most similar to it. The categories of these top k samples are used to determine the categories of each sample. In the classification process, if a sample is more similar to its top k samples, it is more likely to be correctly allocated. As shown in Fig. 1a, t_1 is the sample to be classified, and t_2 – t_7 are the top k samples closest to t_1 in the training set. In KNN, t_1 is represented by t_2 – t_7 in the training set (as shown in Fig. 1b), where it determines the categories of t_1 according to the categories of t_2 – t_7 . In light of this, we think that if we can find a subset that can represent all samples in a training set, it can be used to represent the entire training set. As shown in Fig. 1c, the set S is a subset of the training set T ; if any sample t_m in T can be represented by one or more samples in S , we assume that the set S can represent the entire set T . To obtain set S , we need to find a set from all possible subsets of T that maximizes the sum of similarities among all samples in training set T and their top k samples in S . In this way, almost all training samples can determine nearby samples from the generated subset. In other words, the data distribution of set S is substantially consistent with that of set T . Therefore, samples along the borders of the categories are not neglected. Moreover, as outliers are distant from other samples, they are unlikely to be selected.

However, finding the most suitable set S is an NP-hard problem, for the reason that it involves traversing all subsets of the training set. Thus, an exact solution cannot be obtained within an acceptable time when the training set is large. We propose an optimization algorithm based on the greedy algorithm [9] to obtain an approximate solution that is close to the exact solution.

Several experiments are conducted to compare the performance of our proposed method with baseline sample selection methods. The results show that the AVG-F1 value of Top K representative is higher than those of the density-based method by 3.7% and those of the Uncertainty Sampling by 10.1%, respectively. We also conducted experiments to verify the efficiency of the proposed Top K Representative. The results shows that the run time of our algorithm is acceptable, shorter than 100 s for approximately 10,000 samples.

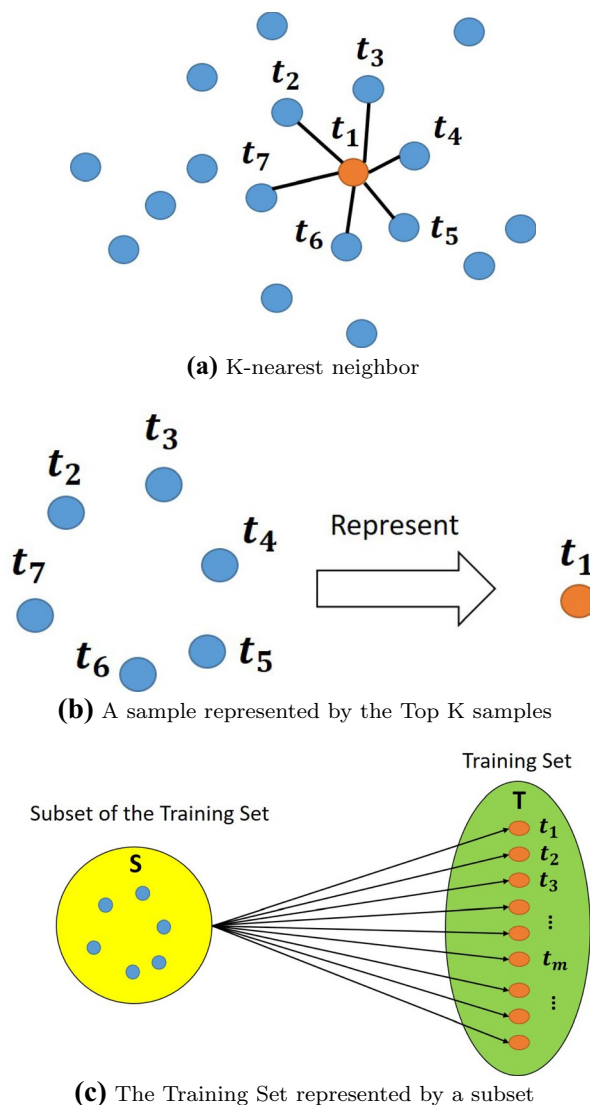


Fig. 1 Idea inspired by KNN. **a** K-nearest neighbor. **b** A sample represented by the Top K samples. **c** The training set represented by a subset

The contributions of our work here can be summarized as follows:

1. We propose a sample selection algorithm, TKR, to determine a subset that can select samples along the borders of categories and avoid selecting outliers.
2. As finding the most suitable subset in TRK is an NP-hard problem, we propose an optimization algorithm based on the greedy algorithm to obtain an approximate solution close to the exact solution.
3. We conducted several experiments to verify the effectiveness of TKR and its efficiency for samples of different sizes.

In the remainder of this paper, we introduce related work in Sect. 2. The proposed model and its corresponding optimization algorithm are detailed in Sect. 3. Our experiments and their results are reported in Sect. 4.

2 Related work

2.1 Active learning

2.1.1 Uncertain sampling

Uncertain sampling [22] is a popular technique of sampling, which is widely-used in nature language processing. For example, Word Sense Disambiguation [5, 6], Text Classification [22, 36], Statistic Parsing [35], Named Entity Recognition [22], image retrieval [17]. Besides, in [18], sentiment classification also apply uncertain sampling approach to find the most informative samples from the corpus in order to reduce human effort.

Uncertain sampling is to choose K samples as initial samples to train the classifier. Then select $N-1$ samples, and use these samples to train classifier $C_N - 1$. Use classifier $C_N - 1$ to classify the samples which have not been selected in data set X , and yield the probability p_{ij} of sample i belong to category j . And then, we compute the entropy:

$$H_i = - \sum_{j=1}^n p_{ij} \log_b p_{ij}, \quad (1)$$

where b is the parameter of the entropy, which usually is 2 or e . In information theory, entropy can measure the uncertainty of information. In our case, if the value of entropy is large, it means the probabilities of the sample belong each class is very close, i.e. it is very hard to decide which class this sample belongs to. Therefore, if the value of entropy is large, we can say the classifier have a poor ability of distinguishing this sample. Putting this sample into the training set is likely to bring more information. So we choose sample:

$$X_N = \operatorname{argmax}_i H_i. \quad (2)$$

The key idea of active-learning based on uncertain sampling is: the sample with the largest uncertainty, which contains the most information, will be selected for manual labeling in every cycle. The uncertainty is large means the current classifier have little confidence of classifying this sample. In this framework, the classifier will choose the samples, which it is uncertain about, to learn. Methods using this framework usually apply probability learning scheme. For example, in the model of binary classification problem, the uncertain sampling just need to select the samples with a probability near 0.5 [22]. In the case of multi-classification, we can use entropy to compute the certainty of classifier. An intuitive

explanation is: uncertain sampling chooses the samples near the decision boundary, and use them to update the decision boundary to get a more precise result. However, if the uncertain sampling chooses outliers, we will fail to get a more precise decision boundary [35]. Although, these outliers have high uncertainties, they can not improve the classifier.

In uncertain sampling, how to choose the initial training set is also very important. In the previous study of active-learning, the initial training set is usually chosen randomly. It is based on the assumption that random sampling might select a set of samples which share the same distribution with the whole data set. However, since we always choose a very small initial training in practice, this assumption might be invalid. In the paper [40], a method which use sampling by cluster (SBC) to select the most typical samples as initial training set. In order to do this, the corpus should be clustered into a specific number (depends on the size of the initial training set) of groups. The samples which are close to the centroids of clusters are the most typical sample.

2.1.2 The selection method based on density distribution

The key idea of the selection method based on density distribution is focusing on the whole input space. Therefore, this method is not likely to choose outliers. It compute the density of the space of samples to choose the typical samples. However, the area with a high density might not be the area near the decision boundary. So we try to combine density with uncertainty.

In the paper [40], a framework about describing the density of information is proposed. It is a general weighting method based on density and uncertainty. Its key idea is that the samples with the most information not only are uncertain, but also locate in an area with a high density in the input space. In [40], the unlabeled samples are evaluated by a density method based on K nearest neighbors. The K nearest neighbors of sample x is represent as the following set:

$$S(x) = \{S_1, S_2, \dots, S_K\}. \quad (3)$$

Hence, the density of the K nearest neighbors $DS(x)$ is defined:

$$DS(x) = \frac{\sum_{s_i} \cos(x, s_i)}{K}, \quad (4)$$

where $\cos(x, s_i)$ is the cosine similarity between sample x and s_i .

There is a new method in [39]. It based on uncertain sampling and density measurement, in which the uncertainty is measured by entropy and the density is measured by K -means method. In this method, we measure the uncertainty using a method called Density * Entropy, which compute as Eq. 5. This is a general method of computing uncertainty and density:

$$DSH(x) = DS(x) * H(x), \quad (5)$$

where $H(x)$ is entropy, which compute through Eq. 1.

Density-based sample selecting approaches have been applied in many research fields. In [18], Density-based sample selecting approaches are introduced into cross-lingual sentiment classification field. Since most recent research works in natural language processing focused on English language, and there are not enough labeled sentiment resources in other languages, sample selecting methods are needed for reducing the cost of manual construction of annotated sentiment corpora for a new language. This work applies density measures of unlabelled samples to avoid outlier selection.

2.1.3 Supervised sampling selecting methods

When active learning combined with some specific applications, supervised sampling selecting methods are introduced, which needs extra labeled samples for training.

In [23], a supervised approach is proposed to solve the domain adaptation problem in sentiment classification. The domain adaptation problem is that a sentiment classifier trained with the labeled samples from one domain has bad performance in another domain. The difficulty of this problem is that the data distributions in the source domains are different from that in the target domains. The authors in [23] perform active learning for cross-domain sentiment classification by a supervised approach which needs a small amount of labeled data for training. First, the labeled source and target data are used to train two individual classifiers. Then, the unlabeled samples will go through the classifiers. The unlabeled samples will be selected when the results of these two classifiers are disagreed.

In [34], a active learning support vector machine (SVM) is proposed for image classification tasks. It prefers to select the uncertain samples from unlabeled dataset as the classifier's training samples. The goal of the active learning task is to select the support vector, which is the most informative samples for classification. In the supervised part, the goal of the labeled samples is to find the best model parameters to obtain a optimal classifier.

The supervised sample selecting methods we discuss above combine active learning and specific applications together. However, these approaches are hard to be applied in other applications. For example, the model proposed in [34] can apply active learning in SVM, but it is hard to be applied in other classifiers like KNN. The model provided in our paper is an unsupervised sample selecting method, and it is a general method developed for applications from different research fields, like text classification [19], image retrieval, sentiment search [13–15, 37], etc.

2.2 The expression of sentiment texts

Most of sentiment texts created from the Internet are the online reviews from E-commerce platforms or articles from Twitter or Facebook. These texts are short in length, thus they are short text. The sentiment classification problem can be transformed to short text classification problem. In short text classification problem, expressing a short text as a vector is an important step. It including two parts: find features to represent the semantic meaning of the text, and the composition of these features. In the field of short text classification, the most widely-used model is Vector Space Model [30].

In Vector Space Model, we usually use Bag of Words model which ignores the order of words, for the reason that samples may be sparse. In this way, a text is expressed as a vector, in which every feature stands for a word, and its value is the weight of the word.

Specific to Chinese, apart from Bag of Words model, N-gram model is also a popular method [4]. It selects N successive words as a phrase(e.g., given a Chinese text:“大学新生” (college freshman), we will have “大学” (college) and “新生” (freshman), if we use Bag of Words; if we use 2-grams model, we will have “大学” (college), “学新” (This combination has no meaning in Chinese), “新生” (freshman), which are the combinations of each pair of successive words.).

2.3 Term weighting

Term weighting is the process of deciding the value of features, when we using Vector Space Model [4, 30] to express a text as a vector. In tradition, term weighting is unsupervised, which means ignoring the label of texts.

There are some popular models:

- 0/1 model: if a word appear in the text, then it will get value 1; if not appear, it get value 0.
- TF-IDF (Term Frequency- Inverse Document Frequency) model [33]: given a text, the Term Frequency is how many times a specific word appear in this text. We usually normalize this value, in case the value become bigger just because the text is longer. Inverse Document Frequency can measure the importance of a word. It is based on the assumption: the less documents a word appears in, the more important this word is.

However, for the reason that the amount of words is too little in a short text, the unsupervised term weighting method cannot perform well. Thus it is better to use supervised methods which take the label of words into account.

IG and χ^2 are two typical statistic measures [29], which can act as the confidence of the dependency between a word

and a category. In these term weighting method, a multi-classification problem is converted into multiple binary classification problems.

3 Top K representative

The solution to the problem of classification of short texts involves a supervised learning process, that needs a dataset labeled by a human. However, creating a massive training corpus is expensive and highly time consuming in practice. Therefore, this process is the bottleneck in the application of such classification to a new field. The purpose of our study is to use active learning to minimize the amount of manual work needed, and to achieve the same final performance as obtained with manual labeling of all data. Active learning is a widely used framework that allows for the selection of samples with the largest amount of information. The key idea underlying active learning is to apply machine learning algorithms to actively acquire training labels from a given dataset [28]. In this manner, we can attain higher accuracy with fewer training labels. We thus attempt to reduce manual labeling work within supervised learning problems through active learning while achieve satisfactory performance.

In this paper, we propose a sample selecting method based on the idea of KNN. The KNN performs well in text classification because the more similar the relevant texts, the more likely they are to belong to the same category [11]. This is exactly how KNN works: given a text, it finds the k most similar samples to it in the training set; it then votes according to the categories of these samples, and chooses the one that receives the most votes [31]. Therefore, if the k samples are strongly similar to the given text, the categorization is probably correct. Conversely, if the samples found are only weakly similar to the text, the categorization is likely incorrect. Given this, we propose a method to select samples, where the idea is to select a subset of the samples for manual labeling. When given a text to test, we can find the top k samples in the subset that are strongly similar to the given text. To this end, we first define the concept of representativeness between samples and subset.

3.1 Representing sample B using sample A

The representativeness of sample A with respect to sample B:

$$R_1(a, b) = \text{Sim}(a, b), \quad (6)$$

where $\text{Sim}(a, b)$ represents the similarity between samples A and B. The representativeness of sample A with respect to sample B is the similarity between A and B. Similarity can be computed using different methods according to the problem at hand. As we focus on short texts in this paper,

we use cosine similarity. Furthermore, according to the tf-idf of the terms appearing in a given text, its feature vector can be represented as:

$$q_i = \langle t_{i1} : w_{i1}, t_{i2} : w_{i2}, \dots, t_{in} : w_{in} \rangle, \quad (7)$$

where t_{ij} are the terms appearing in the data set and w_{ij} is the weight of t_{ij} , where w_{ij} can be computed by tf-idf:

$$w_{ij} = \text{tf}_{ij} * \text{idf}_j = \text{tf}_{ij} * \log\left(\frac{N}{n_j}\right). \quad (8)$$

The definition proposed in this paper is in the context of short text classification, but does not exhaust the meaning of the term. The proposed Top K representative active learning is a general method that can also be used with other types of data than text, e.g., pictures. Depending on the data type, we can use different definitions of representativeness, e.g., the reciprocal of Euclid distance.

3.2 Representing a sample using a set

The representativeness of set S_1 with respect to sample b :

$$R_2(S_1, a) = \sum_{a \in \text{TopKSim}(b, S_1, k)} R_1(S_1, a), \quad (9)$$

where $\text{topKSim}(b, S_1, k)$ represents the k most similar samples to b in S_1 , where k is chosen by a human. $R_2(S_1, a)$ can measure the probability of finding the k most similar samples using test data in set S_1 and the strength of the similarity. We use the sum of the similarity between b and the k samples as the representativeness of set S_1 with respect to the test samples. This is based on the assumption mentioned above: if a set can represent a sample better, the k samples found in the set most similar to this sample must share a stronger similarity with the sample. To measure the k samples at the same time, we consider them equal and take the sum of their similarities to b . This is not the only method to measure these samples. We can alter the method according to the problem. For instance, in some problems, it is important that we find the most similar sample. In this case, we can define $R_2(S_1, a)$ through a weighting method.

3.3 Representing set S_2 using set S_1

The representativeness of set S_1 with respect to set S_2

$$R_3(S_1, S_2) = \sum_{a \in S_1} R_2(S_1, a). \quad (10)$$

Given an unlabeled dataset X , we need to find a subset of X . We call this subset X_S and it should satisfy the following conditions:

- (a) $|X_S| = m$,
 (b) $X_S = \operatorname{argmax}_{X_S \subset X} R_3(X_S, X)$,

where m is the number of samples that we need to label manually. This depends on the number of laborers at hand. Subset X_S , which satisfies the two conditions above, best represents set X . This means that given any sample in set X , we can always find k samples in its subset X_S that share a strong similarity with the sample.

$R_3(X_S, X)$ means the representativeness of a subset with respect to its superset. If this representativeness has a high value, given any sample in the superset, we can find the k samples most similar to the sample in the subset that are strongly similar to it. This means that we have obtained a subset in which we can always find k samples that can satisfactorily represent any sample in the superset, i.e., this subset has a similar distribution to that of its superset.

Algorithm 1 The optimization algorithm

Input: Training set X **Output:** Selected set X_S

```

1 Initialization process
2  $Score(a) = \sum_{b \in X} Sim(a, b)$ 
3  $X_S = \phi$ 
4  $topKSim(a, X_S, k) = \phi$ 
5 while  $|X_S| \neq m$ :
6   do  $a_{selected} = \operatorname{argmax}(Score(a))$ 
7   for  $a \in X$ :
8     do
9       if  $Sim(a_{selected}, a) > KthSim(a, X_S, k)$ :
10        do  $topKSim(a, X_S, k).add(a_{selected})$ 
11      End If
12    if  $|topKSim(a, X_S, k)| > k$ :
13      do //find out the less similar sample in  $topKSim(a, X_S, k)$ 
14         $a_{removed} = LessSimilar(topKSim(a, X_S, k))$ 
15         $topKSim(a, X_S, k).remove(a_{removed})$ 
16      End If
17    if  $|topKSim(a, X_S, k)| = k$ :
18      do  $KthSim(a, X_S, k)_{old} = KthSim(a, X_S, k)$ 
19        //recalculate  $KthSim(a, X_S, k)$ 
20        Recalculate( $KthSim(a, X_S, k)$ )
21      for  $b \in X$ :
22        do
23          if  $Sim(a, b) < KthSim(a, X_S, k)_{old}$ 
24            do continue
25          End If
26          if  $Sim(a, b) > KthSim(a, X_S, k)$ 
27            do  $Score(b) = Score(b) - (KthSim(a, X_S, k) - KthSim(a, X_S, k)_{old})$ 
28            else  $Score(b) = Score(b) - (Sim(a, b) - KthSim(a, X_S, k)_{old})$ 
29            End If
30          End For
31        End If
32      End For
33    End While

```

3.4 Optimization algorithm

We want to find the subset that can best satisfy the aforementioned conditions. Because traversing every subset of a set is NP-hard [20], we cannot obtain a precise result in an acceptable amount of time. However, we do not need a precise solution to this problem, and an approximate result can yield acceptable performance. Therefore, we propose the

greedy algorithm shown in Algorithm 1, which minimizes the size of the subset in each iteration.

Lines 1–4: As X_S is empty at the outset, the $TopKSim$ set of every sample is empty. Therefore, the addition of each sample a to X_S causes the $TopKSim(b, X_S, k)$ of every sample b in dataset X to get a new member a . $R_2(X_S, a)$ then increases $Sim(a, b)$. As $R_3(S1, S2) = \sum_{a \in S1} R_2(S1, a)$, the overall $R_3(X_S, X)$ increases $\sum_{b \in X} Sim(a, b)$. Therefore, $Score(a) = \sum_{b \in X} Sim(a, b)$.

Lines 6–11: Based on the strategy of the greedy algorithm, we choose the sample with the largest value of $Score$ that can increase $R_3(X_S, X)$ by the largest magnitude.

Lines 12–31: Having selected sample $a_{selected}$, if the similarity between $a_{selected}$ and any other sample a is higher than that between it and the k th most similar sample of a , $a_{selected}$ is added to set $topKSim(a, X_S, k)$. At the same time, if the size of set $topKSim(a, X_S, k)$ is greater than k , the member with the lowest similarity in set $topKSim(a, X_S, k)$ should be removed. Following the removal, if the size of set $topKSim(a, X_S, k)$ is k , the $Score$ of other samples may have changed because we may have to remove another sample with the lowest similarity after adding this new sample, b , even if b can eventually stay in the set $topKSim$. Therefore, we must update the $Score$ of every sample in X . We first recompute $KthSim(a, X_S, k)$ and set the old $KthSim(a, X_S, k)$ to $KthSim(a, X_S, k)_{old}$. $KthSim(a, X_S, k) > KthSim(a, X_S, k)_{old}$. If $Sim(a, b) > KthSim(a, X_S, k)$, adding b to X_S first increases $Sim(a, b)$ and then reduces $KthSim(a, X_S, k)$. In this case, $Score(b)$ reduces $KthSim(a, X_S, k) - KthSim(a, X_S, k)_{old}$. Then, if $KthSim(a, X_S, k)_{old} < Sim(a, b) < KthSim(a, X_S, k)$, we remove the part relating to $Sim(a, b)$ from $Score(b)$, which means subtracting $Sim(a, b) - KthSim(a, X_S, k)_{old}$ from it. Otherwise, if $Sim(a, b) < KthSim(a, X_S, k)_{old}$, we do not need to make any change as it is already impossible for b to enter set $TopKSim$.

3.5 Complexity analysis

In our algorithm, when given a new sample, there are two main loops. The first loop checks to determine whether the top k samples need to be updated. When the first loop determines that an update is needed, the second updates the score values of all other samples. Both loops are linearly dependent on $|X|$, and the process of updating the top k samples is linearly dependent on k . The complexity of our algorithm is $O(k * m * |X|^2)$, where m is the number of samples we want to provide for manual labeling and $|X|$ is the size of the entire dataset. For each sample, we need to retain the top k samples. Thus, the spatial complexity is $O(k * |X|)$. To minimize the computation, we use a matrix to record the similarity between each pair of samples. This matrix is sparse (in practice, fewer than 10% of its values are non-zero). We

can thus skip the zero values while traversing the matrix to improve the speed of computation.

4 Experiments

4.1 Datasets

In this experiment, we compare the performance of different sampling methods. The proposed sampling selecting method is a general method and can be applied in different field, for example, short text classification or sentimental classification. In the experiments, we apply the sample selecting methods in short text classification. Samples are selected using different sampling methods, and they are used to train classifiers. The performance of classifiers represents the performance of the corresponding sampling method. In this experiment, we assume that the dataset does not have any labels. A sample will obtain its label when it is selected by the sampling methods. We apply Google Snippets dataset [26] in our experiment. This dataset consists of 10,060 training snippets and 2280 test snippets from 8 categories. On average, each snippet has 18.07 words. 10% of this dataset is used as the test set, denoted as T. Meanwhile, 90% of that is regarded as a optional set, denoted as X. We do not use all the sample in optional set X as our training set. Instead, we only use N sample of X as the training set. Ten-Fold cross-validation [24] is applied to make the result convincible.

4.2 Evaluation metric

The evaluation matric used in this paper is AVG-F1 [27]. It is calculated as follows:

$$AVG - F1 = \frac{1}{N} \sum_{i=1}^N F1_i, \tag{11}$$

where N is the number of classifiers, and $F1_i$ is the $F1$ value for classifier i . $AVG - F1$ is the average $F1$ value of all classifiers. $F1_i$ is calculated as follows [27]:

$$F1_i = 2 * \frac{Precision_i * Recall_i}{Precision_i + Recall_i}. \tag{12}$$

The precision and recall value is calculated as follows [12, 25]:

$$Precision_i = \frac{TP_i}{TP_i + FP_i}, \tag{13}$$

$$Recall_i = \frac{TP_i}{TP_i + FN_i}, \tag{14}$$

where TP_i is the number of samples correctly classified as belonging to the positive class by classifier i , and FP_i is the number of samples that is incorrectly classified to positive class, while FN_i is the number of samples incorrectly classified as belonging to the negative class.

4.3 Comparing methods

The comparing sampling methods in this experiment is introduced as follows:

- Random sampling: N samples are selected randomly as the training set [7].
- Uncertainty Sampling: K samples are selected randomly at first. And then another N samples are selected one by one. When selecting Nth sample, the selected N-1 samples are used to train a classifier C_{N-1} . This classifier can obtain the probability p_{ij} that sample i belong to class j . Then the entropy of sample i can be calculated in Eq. 1. In information theory [32], entropy is used to measure the uncertainty. Higher entropy value represents that the probability of a sample belonging to each class is similar, thus it is harder to determine a class for that sample. Therefore, the high entropy of a sample indicates that the classifier has a poor performance on this sample, thus this sample should be put into the training set to bring more classifying information to the classifier. The Nth sample should be selected in Eq. 2.
- Density*Entropy: Zhu et al. propose a method to describe the density of information [40], called Density*Entropy. This is a weighted method based on density and uncertainty. Its main idea is that the most informative sample is not only those with uncertainty, but also those in the dense area in the input space. In [40], the unlabeled samples are evaluated by a density method based on K

Table 1 AGV-F1 value of different sampling methods

Sample size	Random	Uncertainty sampling	Density * Entropy	Top K representative
50	0.2796	0.1165	0.2998	0.3240
100	0.4235	0.2352	0.4576	0.4740
150	0.3885	0.2710	0.4792	0.4830
200	0.4714	0.3038	0.4979	0.5223
250	0.4786	0.3872	0.5576	0.5898
300	0.5218	0.4144	0.5877	0.5823
350	0.5387	0.4486	0.5862	0.5985
400	0.5608	0.4744	0.5636	0.5815
450	0.5703	0.5087	0.5781	0.6054
500	0.5867	0.5061	0.5998	0.6135

Fig. 2 The performance of different sampling methods

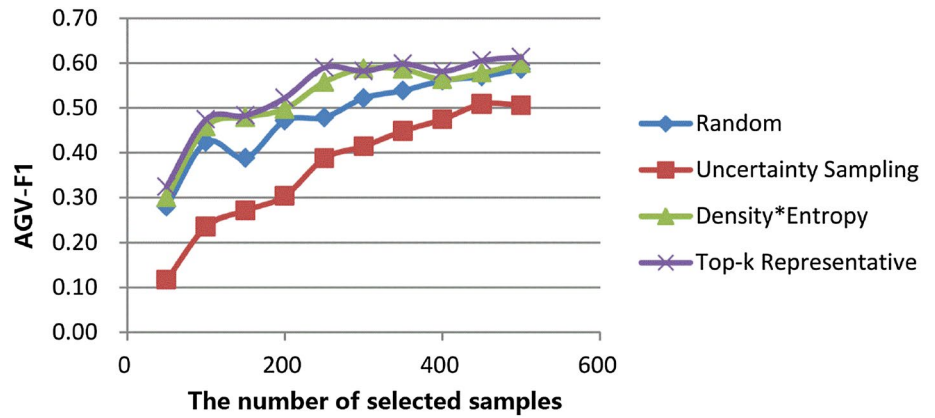


Table 2 Relation between the number of samples in optional set $|X|$ and the run time

The number of samples in optional set (X)	The running time (s)
200	0.147
400	0.342
600	0.544
800	0.867
1000	1.286
2000	3.817
4000	13.267

4.4 Comparison of sample selection methods

We also compared the performance of the proposed TKR with other samples selection methods. We chose random sampling, uncertainty sampling and Density * Entropy as the baseline methods, as they are popular sample selection techniques. These methods were applied to select samples from the Snippets dataset, and the selected samples were used to train a KNN classifier. We used AVG-F1 value [27] to evaluate the performance of each sampling method according to the performance of the corresponding classifier. The experimental results are shown in Fig. 2 and Table 1. As shown in the graph in the figure, classification accuracy increased with the number of selected samples because with more samples, the classifiers became better informed and could identify more features in the dataset. The figure also shows that our proposed Top K Representative yielded a higher AVG-F1 value than the other methods when selecting the same number of samples. The gap between Top K Representative and the other methods narrowed with increase in the number of selected samples because the corresponding classifiers then had sufficient information to more accurately choose samples. We also found that the AVG-F1 values of Density * Entropy and TKR were similar and higher than

nearest neighbors. The K nearest neighbors of sample x is represent as the following set: $S(x) = \{S_1, S_2, \dots, S_k\}$. Hence, the density of the K nearest neighbors $DS(x)$ is defined in Eq. 4. The formula of Density*Entropy is in Eq. 5. This method is widely used to combine uncertainty sampling methods and density-based sampling methods together.

- Top K representative: the sampling method we proposed in this paper.

Fig. 3 Relation between the number of samples in optional set $|X|$ and the run time

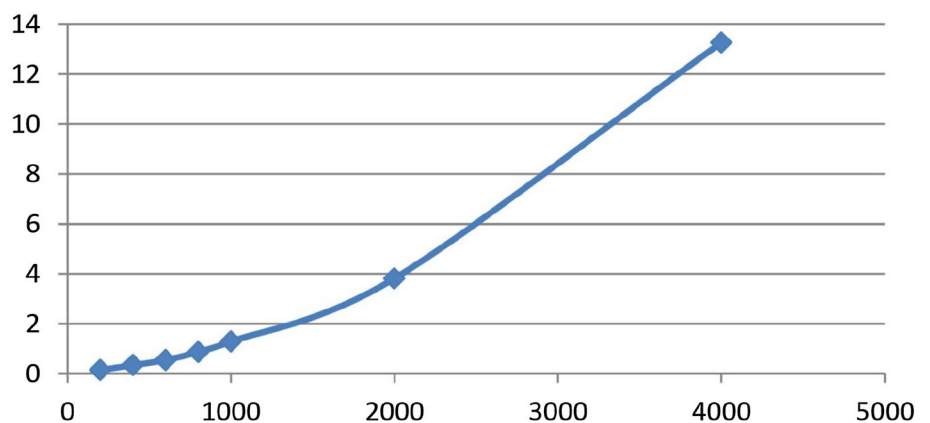


Table 3 Relation between the number of samples we selected m and the run time

The number of target samples (m)	The running time (s)
100	19.221
200	20.823
300	22.416
400	23.211
500	24.209
800	26.176
1000	26.863
1500	30.166
2000	31.664
3000	34.626

those of random sampling and uncertainty sampling. This is because the latter methods struggle to solve problems featuring outliers, whereas TKR and Density * Entropy can handle them. The AVG-F1 value of TKR was higher than that of Density * Entropy, the second-best method, by 3.7% because the former could select important samples along the boundary of the categories whereas the latter could not. Note that the AVG-F1 value of Top K representative for 100 selected samples was 0.47, whereas random sampling attained this value for 200 samples, and uncertainty sampling needed 400 samples obtain the same value. That is to say that our proposed method can reduce labeling work by half while delivering the same performance as the other methods. This shows that our proposed method can save both the time and human resources needed to label samples.

In general, such traditional methods as uncertainty sampling perform worse than random selecting on short texts. We found that uncertainty sampling tended to select more outliers in a database of short texts. This had a negative effect on classification performance because the short text dataset is usually a sparse dataset; that is, some words in the samples have never appeared before. These samples are

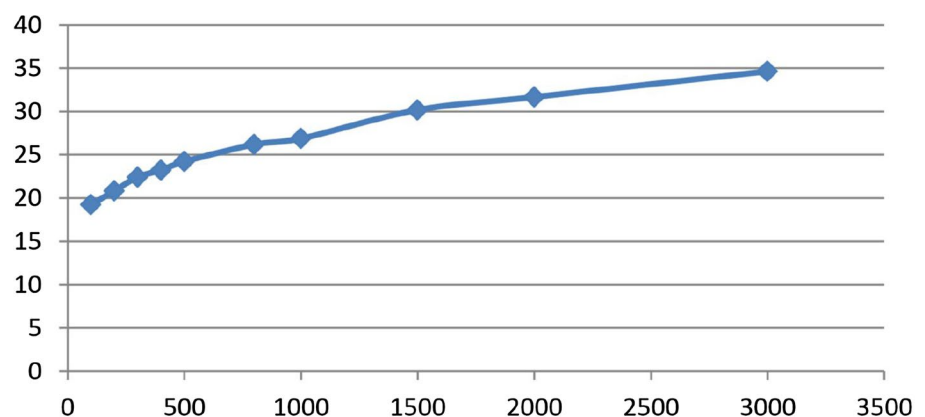
at large distances from others, and classifiers cannot obtain the information needed from the other samples to classify them. According to uncertainty sampling, these samples are selected as training samples and make a small contribution to the classification process. Thus, uncertainty sampling yields poor performance on short texts. However, our proposed method and Density * Entropy do not encounter this problem because they choose samples that can represent the entire dataset. As outliers have low similarity with other samples in a given dataset, the samples obtained by our proposed method are less likely to be outliers.

4.5 Time efficiency of top K representative

This part of our experiment was intended to verify the efficiency of the optimization algorithm proposed in this paper and gauge its complexity. We also used the Snippets dataset in this experiment. There are two parameters in the optimization algorithm that can influence efficiency. The first is the total number of samples, i.e., the number of samples in the optional set, denoted by $|X|$, and the second parameter is the number of samples we want to select from all samples, denoted by m .

The relation between the number of samples in optional set $|X|$ and the run time is shown in Table 2 and Fig. 3. We fixed m to 100, i.e., $m = 100$. As show in the above results, the run time was close to the quadratic of $|X|$, which increased 2.96 times from 1000 to 2000, and 3.47 times from 2000 to 4000. The quadratic relation was not obvious when the value of $|X|$ was low because program initialization and other factors occupied part of the necessary time. However, with increase in the number of samples, the common part [Common to/between what? Please specify.] of the run time decreased, because of which the curve of $|X|$ with respect to s was close to the quadratic function.

We conducted another experiment to explore the relationship between the number of target samples m and run time. The results are shown in Table 3 and Fig. 4. We fixed the

Fig. 4 The relation between the number of samples we selected m and the running time

number of samples in the optional set to $|X| = 12387$. The figure shows that run time s and the number of target samples m had a linear relationship over time. This is because the similarities among all samples were calculated in the initialization process, which can consume a large amount of time even if the number of target samples is small. For example, it took approximately 19.221 s to select only 100 samples because most of the time was consumed by initialization. The run time increased to 34.626 s when the number of target samples was increased to 30,000. The linear relationship was apparent when the number of target samples was greater than 100. In this case, the run time increased by 1–2 s every 100 samples.

In general, it is clear from the experiment that the complexity of the optimization algorithm is $O(k * m * |X|^2)$. The run time of our algorithm for a sample size of approximately 10,000 was less than 100 s, an acceptable result. Thus the proposed algorithm can be applied to practical problems.

5 Conclusion and future work

In this paper, we propose a method of selecting samples for manual labeling. The proposed method can be applied in different fields, for example, text classification, sentiment classification and so on to reduce the labeling workload of datasets. Firstly, we propose a series of definitions of representativeness. And then we convert the selection problem to find the subset which have the highest representativeness. As this is an NP-hard problem, and we do not need a precise result, we propose a greedy algorithm to solve it.

We did not consider updating the model following manual labeling. Therefore, sample selection was unsupervised. Further work can be done here by using manually labeled samples to improve the model, or to combine them with uncertainty sampling to implement a method for distribution estimation and uncertainty estimation. Although the greedy algorithm we used is efficient, the final result is not a precise answer. Further research in the area can use the hill-climbing algorithm, the genetic algorithm, or other algorithms to yield better results and improve sampling performance.

Acknowledgements This work is supported by the Fundamental Research Funds for the Central Universities, SCUT (No. 2017ZD048), Tiptop Scientific and Technical Innovative Youth Talents of Guangdong special support program (No. 2015TQ01X633), Science and Technology Planning Project of Guangdong Province, China (No. 2017B050506004), Science and Technology Program of Guangzhou (International Science & Technology Cooperation Program No. 201704030076), and the Internal Research Grant (RG 66/2016-2017) and the Funding Support to ECS Proposal (RG 23/2017-2018R) of The Education University of Hong Kong.

References

1. Aas K, Eikvil L (1999) Text categorisation: A survey. vol 167. Technical report, Norwegian computing center, p 306
2. Aggarwal CC, Zhai CX (2012) Mining text data. Springer, Boston, MA
3. Altman NS (1992) An introduction to kernel and nearest-neighbor nonparametric regression. *Am Stat* 46(3):175–185
4. Broder AZ, Glassman SC, Manasse MS, Zweig G (1997) Syntactic clustering of the web. *Comput Netw ISDN Syst* 29(8):1157–1166
5. Chan YS, Ng HT (2007) Domain adaptation with active learning for word sense disambiguation. In: Annual Meeting-association for computational linguistics, vol 45, p 49
6. Chen J, Schein A, Ungar L, Palmer M (2006) An empirical study of the behavior of active learning for word sense disambiguation. In Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics, pp 120–127. Association for Computational Linguistics
7. Cochran WG (2007) Sampling techniques. 3rd edn. Wiley, New York
8. Coomans D, Massart DL (1982) Alternative k-nearest neighbour rules in supervised pattern recognition: part 1. k-nearest neighbour classification by using alternative voting rules. *Analytica Chimica Acta* 136:15–27
9. Cormen Thomas H, Charles Eric Leiserson, Ronald L Rivest, Clifford Stein (2001) Introduction to algorithms. MIT Press, Cambridge
10. Dai HK, Zhao L, Nie Z, Wen J-R, Wang L, Li Y (2006) Detecting online commercial intention (oci). In: Proceedings of the 15th international conference on World Wide Web, pp 829–837. ACM
11. Everitt BS, Landau S, Leese M, Stahl D (2011) Miscellaneous clustering methods. In: Cluster Analysis, 5th edn. Wiley, New York, p 215–255
12. Fawcett T (2006) An introduction to roc analysis. *Pattern Recognit Lett* 27(8):861–874
13. Fu Z, Huang F, Sun X, Vasilakos A, Yang C (2017) Enabling semantic search based on conceptual graphs over encrypted outsourced data. *IEEE Trans Serv Comput* 99:1–1
14. Zhangjie F, Ren K, Shu J, Sun X, Huang F (2016) Enabling personalized search over encrypted outsourced data with efficiency improvement. *IEEE Trans Parallel Distrib Syst* 27(9):2546–2559
15. Zhangjie Fu, Xinle Wu, Guan Chaowen, Sun Xingming, Ren Kui (2016) Toward efficient multi-keyword fuzzy search over encrypted outsourced data with accuracy improvement. *IEEE Trans Inf Forensics Secur* 11(12):2706–2716
16. Gabrilovich Evgeniy, Markovitch Shaul (2007) Computing semantic relatedness using wikipedia-based explicit semantic analysis. *IJCAI* 7:1606–1611
17. *IEEE Trans Image Process* (2008) Active learning methods for interactive image retrieval. 17(7):1200–1211
18. Hajmohammadi MS, Ibrahim R, Selamat A, Fujita H (2015) Combination of active learning and self-training for cross-lingual sentiment classification with density analysis of unlabelled samples. *Inf Sci* 317:67–77
19. Hoi SCH, Jin R, Lyu MR (2006) Large-scale text categorization by batch mode active learning. In: Proceedings of the 15th international conference on World Wide Web, pp 633–642. ACM
20. Knuth DE (1974) Postscript about np-hard problems. *ACM SIGACT News* 6(2):15–16
21. Lewis DD, Catlett J (1994) Heterogeneous uncertainty sampling for supervised learning. In: Proceedings of the eleventh international conference on machine learning, pp 148–156
22. Lewis DD, Gale WA (1994) A sequential algorithm for training text classifiers. In: Proceedings of the 17th annual international

- ACM SIGIR conference on Research and development in information retrieval, pp 3–12. Springer, New York
23. Li S, Xue Y, Wang Z, Zhou G (2013) Active learning for cross-domain sentiment classification. In: IJCAI, pp 2127–2133
 24. McLachlan GJ, Do KA, Ambrose C (2004) Analyzing microarray gene expression data. Wiley, New York
 25. Perruchet Pierre, Peereman Ronald (2004) The exploitation of distributional information in syllable processing. *J Neurol* 17(2):97–119
 26. Phan X-H, Nguyen L-M, Horiguchi S (2008) Learning to classify short and sparse text and web with hidden topics from large-scale data collections. In: Proceedings of the 17th international conference on World Wide Web, pp 91–100. ACM
 27. Powers DM (2011) Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. 2(1):37–63
 28. Prince Michael (2004) Does active learning work? A review of the research. *J Eng Educ* 93(3):223–231
 29. Salton G, McGill MJ (1986) Introduction to modern information retrieval. McGraw-Hill Inc., New York
 30. Salton G, Wong A, Yang C-S (1975) A vector space model for automatic indexing. *Commun ACM* 18(11):613–620
 31. Samworth RJ et al (2012) Optimal weighted nearest neighbour classifiers. *Ann Stat* 40(5):2733–2763
 32. Shannon CE (2001) A mathematical theory of communication. *ACM SIGMOBILE Mob Comput Commun Rev* 5(1):3–55
 33. J Doc (1972) A statistical interpretation of term specificity and its application in retrieval. 28(1):11–21
 34. Sun F, Yan X, Zhou J (2016) Active learning svm with regularization path for image classification. *Multimedia Tools Appl* 75(3):1427–1442
 35. Min T, Xiaoqiang L, Salim R (2002) Active learning for statistical natural language parsing. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, pp 120–127. Association for Computational Linguistics
 36. Wang JZH, Hovy E (2008) Learning a stopping criterion for active learning for word sense disambiguation and text classification. In: Third International Joint Conference on Natural Language Processing, p 366
 37. Xia Z, Wang X, Sun X, Wang Q (2016) A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data. *IEEE Trans Parallel Distrib Syst* 27(2):340–352
 38. Yang K, Cai Y, Cai Z, Tan X, Xie H, Wong TL, Chan WH (2017) A new samples selecting method based on k nearest neighbors. In *Big Data and Smart Computing (BigComp)*, 2017 IEEE International Conference on IEEE, pp 457–462
 39. Zhu J, Wang H, Tsou BK (2010) Active learning with sampling by uncertainty and density for data annotations. *IEEE Trans Audio Speech Lang Process* 18(6):1323–1331
 40. Zhu Ji Wang H, Yao T, Tsou BK (2008) Active learning with sampling by uncertainty and density for word sense disambiguation and text classification. In: Proceedings of the 22nd International Conference on Computational Linguistics vol 1, pp 1137–1144. Association for Computational Linguistics