Active Learning from Crowds with Unsure Option

Jinhong Zhong[1], **Ke Tang**[1*] **and Zhi-Hua Zhou**[2]
[1]UBRI, School of Computer Science and Technology
University of Science and Technology of China, Hefei 230027, China
[2]National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China
jinhong@mail.ustc.edu.cn, ketang@ustc.edu.cn, zhouzh@lamda.nju.edu.cn

Abstract

Learning from crowds, where the labels of data instances are collected using a crowdsourcing way, has attracted much attention during the past few years. In contrast to a typical crowdsourcing setting where all data instances are assigned to annotators for labeling, *active learning from crowds* actively selects a subset of data instances and assigns them to the annotators, thereby reducing the cost of labeling. This paper goes a step further. Rather than assume all annotators must provide labels, we allow the annotators to express that they are unsure about the assigned data instances. By adding the "unsure" option, the workloads for the annotators are somewhat reduced, because saying "unsure" will be easier than trying to provide a crisp label for some difficult data instances. Moreover, it is safer to use "unsure" feedback than to use labels from reluctant annotators because the latter has more chance to be misleading. Furthermore, different annotators may experience difficulty in different data instances, and thus the unsure option provides a valuable ingredient for modeling crowds' expertise. We propose the ALCU-SVM algorithm for this new learning problem. Experimental studies on simulated and real crowdsourcing data show that, by exploiting the unsure option, ALCU-SVM achieves very promising performance.

1 Introduction

It is often very costly (e.g., time consuming) to label an instance in a real-world application. By carefully choosing a subset of instances to label, active learning is expected to result in a good learner with less labeling cost than traditional learning techniques. Specifically, such a subset is chosen iteratively with query heuristics, which may query instances that are informative [Settles *et al.*, 2008], representative [Nguyen and Smeulders, 2004], or both [Huang *et al.*, 2010].

Existing active learning approaches can be categorized as using a single annotator [Roy and McCallum, 2001] or using multiple annotators [Dekel *et al.*, 2012]. The former is regarded as traditional active learning and the latter, which is essentially a kind of ensemble method [Zhou, 2012], is often referred to as Active Learning from Crowds (ALC) and is attracting increasing research interests thanks to the development of crowdsourcing techniques, e.g., the Amazon Mechanical Turk platform.

Since multiple annotators are involved, an ALC algorithm needs to identify not only the instances to query, but also the appropriate annotators for the chosen instances. Yet choosing annotators is a non-trivial task, as it is unknown in advance what instances an annotator will label correctly. Hence, estimating the expertise of candidate annotators plays a key role in ALC algorithms. So far, much effort has been made toward this purpose [Long *et al.*, 2013] [Yan *et al.*, 2011]. However, all the existing methods try to address this challenging task in a passive way, assuming all annotators must provide labels. This paper goes a step further, the annotators are allowed to express that they are unsure about the assigned instances. The "unsure" feedback is safer than the labels from reluctant annotators because the latter has more chance to be misleading. Moreover, different annotators may feel difficulty in different instances, and thus the unsure option provides a valuable ingredient for modeling crowds' expertise.

Motivated by this consideration, a variant of ALC, namely *Active Learning from Crowds with Unsure option* (ALCU), is formulated and investigated in this work. In ALCU, an annotator not only may provide either labels to queried instances (just like in standard ALC), but also may express that he/she is unsure about the instances' labels. Specifically, the contributions of this work include:

1). The ALCU problem is formulated.

2). A Support Vector Machine (SVM) approach, namely ALCU-SVM, is proposed to tackle the ALCU problem.

3). Empirical studies on both simulated and real crowdsourcing data are conducted, which demonstrate that ALCU-SVM can significantly enhance the effectiveness of ALC. Thus, the importance of providing the unsure option in ALC is justified.

The rest of the paper is organized as follows. The related work of ALCU is reviewed in Section 2. In Section 3, the problem description of ALC and ALCU are presented. The analysis of ALCU and the proposed approach ALCU-SVM

*Corresponding author

are introduced in Section 4. Section 5 presents experimental studies. Finally, the paper concludes with some discussions in Section 6.

2 Related Work

As mentioned above, one of the most important issues of learning from crowds is to model the expertise of annotators. According to different assumptions they offer regarding annotators' expertise, existing works on ALC can be categorized into 3 groups:

First, in their pioneering work, Sheng and Provost [Sheng *et al.*, 2008] assume that an annotator gives correct labels with a certain probability, and the probability is assumed to be the same for all annotators in all instances. Albeit overly simplified, this assumption facilitates understanding some of the most fundamental issues of ALC.

The second type of ALC methods [Long *et al.*, 2013] [Donmez *et al.*, 2009] [Zhao *et al.*, 2014] does not assume all annotators will behave exactly the same. Instead, the probability that an annotator provides the correct label of an instance is characterized as a function of two factors, namely the reliability of the annotator and the difficulty of the instance. The reliability of an annotator is assumed to be the same over all instances, but reliability of one annotator may differ from that of another. In [Donmez *et al.*, 2009] and [Long *et al.*, 2013], all instances are assumed to be of the same difficulty, and different methods are proposed to estimate the reliability of annotators. Zhao et al. [Zhao *et al.*, 2014] go one step further by introducing different difficulties of instances and developing an approach to estimate both reliability of annotators and difficulties of instances.

If the reliability of an annotator is assumed to be the same over all instances, annotators with higher reliability will be more likely to be chosen for all the instances. This may lead to choosing inappropriate annotators. Alternatively, the reliability of annotators could also be formulated as joint probabilistic models of instances and annotators [Yan *et al.*, 2011] [Yan *et al.*, 2012]. In each iteration, the joint probabilistic models are first estimated, and then the instance to query as well as the optimal annotator are chosen simultaneously based on the models. This type of methods directly estimates the probability that each annotator provides the correct label for each instance and thus does not suffer from the drawbacks induced by the assumptions made in the previous two categories. On the other hand, it involves a more challenging intermediate task, i.e., estimating the accuracy of each annotator in each instance, which is non-trivial to solve.

In addition to the research on ALC, it is also noteworthy that the term "unsure option" has also been considered in research on abstaining classifiers [Kwok, 1999] [Pietraszek, 2005] [Friedel *et al.*, 2006]. However, these works concern learning a classifier that can output an "unsure" label. In ALCU, the learned classifier is not allowed to output such a label. Instead, it is the input of a learning algorithm (i.e., training data) that may consist of "unsure" labels. Thus, albeit relevant to each other, ALCU and training abstaining classifiers actually consider different learning problems.

3 Problem Description and Analyses

Let $\{\mathbf{x}_1, \mathbf{x}_2, \dots \mathbf{x}_N\}$ denote N instances and T denote the number of annotators. Without loss of generality, a binary classification problem is considered in this work. The correct label of the j-th instance is denoted by z_j, i.e., $z_j = \{+1, -1\}$. Let y_j^t denote the label of annotator t on the j-th instance. By providing annotators the unsure option, y_j^t can be +1, -1 or 0 in ALCU, where 0 indicates that the annotator is unsure in that instance. It should be noted that in a real-world ALCU scenario, a zero y_j^t is substantially different from the case that the value of y_j^t is missing, because the reason for the latter case might be multi-folds rather than that the annotator is unsure about the label. Since ALCU is a variant of ALC with a new option for annotators' feedback, a unified view that we take on ALC is first described below. The proposed algorithm for tackling ALCU will be detailed in the next section.

We denote the target classifier to be achieved by active learning as $f(\mathbf{x})$ and the classifier obtained after i rounds of queries as $f_i(\mathbf{x})$. An active learning algorithm relies on the query heuristic to identify the instance $\mathbf{x}$ to query in each iteration. For many commonly-used active learning algorithms (e.g., [Lewis and Catlett, 1994] [Dagan and Engelson, 1995]), the query heuristics can be viewed as maximizing some reward functions that take the form $H(\mathbf{x}|\mathbb{L}_i, f_i, \theta_i)(H > 0)$, i.e., the queried instances is picked in the $(i+1)$-th iteration according to Eq. (1):

$$\mathbf{x}_{i+1} = \arg\max_{\mathbf{x}\in\mathbb{U}_i} H(\mathbf{x}|\mathbb{L}_i, f_i, \theta_i) \tag{1}$$

where $\mathbb{U}_i$ is the current set of unlabeled instances, $\mathbb{L}_i$ is the current labeled data, and θ_i is the parameter of H.

In the context of ALC, a query would be beneficial only if a correct label is obtained. We further introduce $g_t(\mathbf{x})$, a function over $\mathbf{x}$, to denote the true reliability of annotator t. Also, a function of $g_t(\mathbf{x})$, denoted by $\Omega(\mathbf{x})$, is used to represent the reliability of the crowds (i.e., the set of all annotators). Following this viewpoint, Eq. (1) can be re-written as Eq. (2) for ALC.

$$\mathbf{x}_{i+1} = \operatorname*{argmax}_{\mathbf{x}\in\mathbb{U}_i}(H(\mathbf{x}|\mathbb{L}_i, f_i, \theta_i) \cdot \Omega(\mathbf{x})) \tag{2}$$

The advantage of viewing ALC as Eq. (2) is that the selection of annotator is not involved at this stage. That is, the selection of queried instance and annotator can be tackled separately. On the other hand, the challenge is that $\Omega(\mathbf{x})$ needs to be specified such that it can be calculated with $g_t(\mathbf{x})$. One way to address this issue is to take a binary view on $\Omega(\mathbf{x})$ and $g_t(\mathbf{x})$. Concretely, $\Omega(\mathbf{x})$ = 1 if $\mathbf{x}$ can be correctly labeled by at least one annotator in the crowds and $\Omega(\mathbf{x}) = 0$ otherwise. Similarly, let $g_t(\mathbf{x}) > 0$ if annotator t can label $\mathbf{x}$ correctly and otherwise $g_t(\mathbf{x}) \leq 0$. Finally, Eq. (3) can be obtained.

$$\Omega(\mathbf{x}) = \vee_{t\in\mathbb{T}}\delta(g_t(\mathbf{x})) \tag{3}$$

where $\vee$ is the disjunction function, for any x_1, $x_2 \in \{0, 1\}, x_1 \vee x_2 = 0$ if and only if $x_1 = 0$ and $x_2 = 0$, $x_1 \vee x_2 = 1$ otherwise. $\delta(x)$ is the sign function, $\delta(x) = 1$ if $x > 0$ and $\delta(x) = 0$ otherwise. $\mathbb{T} = \{1, 2, \dots, T\}$.

Eq. (3) requires $g_t(\mathbf{x})$ to be known. In practice, however, it is unlikely that $g_t(\mathbf{x})$ is known in advance. Alternatively, it can be iteratively learned from the responses of annotators. When learning $g_t(\mathbf{x})$, an instance is treated as negative if the t-th annotator chooses the unsure option, and positive otherwise. When a response is received from annotator t, $g_t(\mathbf{x})$ is updated and then $\Omega(\mathbf{x})$ is recalculated with the updated $g_t(\mathbf{x})$.

For an instance $\mathbf{x}$ selected using Eq. (2), an annotator with $g_t(\mathbf{x}) > 0$ can be queried. If multiple such annotators exist, the one with the largest $g_t(\mathbf{x})$ is chosen, i.e., with Eq. (4)

$$t_{i+1} = \underset{t \in \mathbb{T}}{\operatorname{argmax}}\, g_t(\mathbf{x}_{i+1}) \tag{4}$$

The above idea builds up a general framework for tackling ALCU. Albeit simple, it might not be the optimal one. Other potential approaches are discussed in Section 6.

4 ALCU-SVM

The basic idea of ALCU-SVM is modeling $g_t(\mathbf{x})$ as a binary classifier. We assume in ALCU-SVM that the target classifier and reliability model of annotators take the following forms:

$$f(\mathbf{x}) = \sum_k \alpha_k K(\mathbf{x}_k, \mathbf{x}) + b \tag{5}$$

$$g_t(\mathbf{x}) = \sum_k \beta_{t,k} K_t(\mathbf{x}_k, \mathbf{x}) + c_t \tag{6}$$

where $K(\cdot,\cdot)$ and $K_t(\cdot,\cdot)$ are the kernel functions of target classifier and annotator t's reliability model respectively.

Further, ALCU-SVM employs uncertainty sampling [Settles, 2010] as the query strategy. The reward function used in this paper can be defined as follows:

$$H(\mathbf{x}|\mathbb{L}_i, f_i, \theta_i) = P(z=1|\mathbf{x}, f_i)(1 - P(z=1|\mathbf{x}, f_i)) \tag{7}$$

where $P(z=1|\mathbf{x}, f_i)$ is estimated using Eq. (8):

$$P(z=1|\mathbf{x}, f_i) = (1 + exp(-f_i(\mathbf{x})))^{-1} \tag{8}$$

Since maximizing (7) is equivalent to minimizing $|f_i(\mathbf{x})|$ [Yan *et al.*, 2011], Eqs. (2) and (4) can be rewritten as Eqs. (9)-(10), respectively.

$$\mathbf{x}_{i+1} = \underset{\mathbf{x} \in \mathbb{U}_i}{\operatorname{argmin}} (\sum_k \alpha_k K(\mathbf{x}_k, \mathbf{x}) + b)^2 + C(1 - \Omega(\mathbf{x})) \tag{9}$$

$$t_{i+1} = \underset{t \in \mathbb{T}}{\operatorname{argmax}}\, g_t(\mathbf{x}_{i+1}) \tag{10}$$

where C is a sufficiently large constant that all the instances $\mathbf{x}$, which is subject to $\Omega(\mathbf{x}) \neq 1$, would not be selected.

It is note-worthy that Eq. (9) explicitly biases toward the instances in which at least one annotator is deemed to be sufficiently reliable, i.e., $\Omega(\mathbf{x}) = 1$. However, the reliability models of annotators might not be accurate enough, especially in the early stages of learning. Hence, it is possible that the most informative instance with respect to Eq. (1), would not be chosen to be queried even if there exist some annotators who can label it correctly. To overcome this problem, the most informative instance $\mathbf{x}'$ is chosen using Eq. (11)(maximizing (1) is equivalent to minimizing (11) for ALCU-SVM), and if $\mathbf{x}'$ is different from the instance chosen using Eq. (9),

Algorithm 1: ALCU-SVM

Data: Max number of queries Num, initial data $\mathbf{X}_0$, initial observed label $\mathbf{Y}_0$, unlabeled data $\mathbf{X}_u$
Result: classifier f, reliability models of annotators $\{g_t\}$
begin

```
1  Initialize f and {g_t} with X_0 and Y_0
2  i = -1
3  while i < Num - 1 do
4      i = i + 1
5      Using Eq. (11) to get the x'
6      Using Eq. (9) and (10) to get x_{i+1} and t_{i+1}
7      Query the label of x_{i+1} from t_{i+1}
8      Remove x_{i+1} from unlabeled set
9      Update f, w_{t_{i+1}} and g_{t_{i+1}}
10     if x_{i+1} ≠ x' & i < Num - 1 then
11         i = i + 1
12         x_{i+1} = x'
13         Using Eq. (10) to get annotator t_{i+1} to label
           x_{i+1}
14         Query the label of x_{i+1} from t_{i+1}
15         Remove x_{i+1} from unlabeled set
16         Update f , w_{t_{i+1}} and g_{t_{i+1}}
```

it is also queried with the most reliable annotator identified with Eq. (10).

$$\mathbf{x}_{i+1} = \underset{\mathbf{x} \in \mathbb{U}_i}{\operatorname{argmin}} (\sum_k \alpha_k K(\mathbf{x}_k, \mathbf{x}) + b)^2 \tag{11}$$

Building the reliability model for an annotator may be a class-imbalanced problem since the positive/negative data collected for a highly unskilled/skilled annotator will be rare. Quite a few existing techniques [Batuwita and Palade, 2010] [He and Garcia, 2009] can be adopted to address this problem. For the sake of simplicity, training instances are assigned with weights proportional to the sizes of class in the whole training data, i.e.:

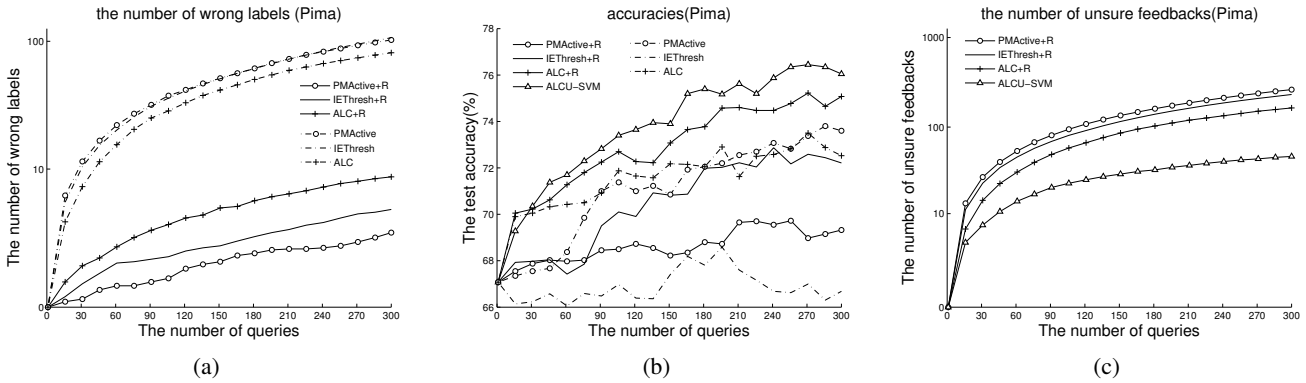
$$w_{p,t} : w_{n,t} = N_{n,t} : N_{p,t} \tag{12}$$

where $w_{p,t}$ and $w_{n,t}$ are the weights assigned to positive data (labeled data) and negative data (unsure data) when training $g_t(\mathbf{x})$. $N_{p,t}$ and $N_{n,t}$ are the sizes of positive and negative data, respectively, collected for annotator t.

Algorithm 1 outlines the steps of ALCU-SVM, where $\mathbf{w}_t$ denotes $[w_{p,t}, w_{n,t}]$. In line 10 of Algorithm 1, $\mathbf{x}_{i+1} \neq \mathbf{x}'$ indicates that the current reliability models cannot suggest a sufficiently reliable annotator to label the most informative instance $\mathbf{x}'$. However, there may be some annotators who can label it correctly while the current reliability models are not accurate enough in this instance. As the classifier can greatly benefit from obtaining the correct label of $\mathbf{x}'$, it is necessary to obtain this label by all means. This is what lines 12-16 do.

5 Experiments

Empirical studies on three UCI data sets (including Pima, Heart and Ionosphere) and a real crowdsourcing data set have



(a) (b) (c)

Figure 1: The comparison of different methods on Pima datasets

been conducted. Overall, through these experiments, the following questions will be investigated:

(a). The benefit of providing the annotators an unsure option.

(b). The advantages of ALCU-SVM over other ALC methods for ALCU problems.

(c). The influence of reliability model selection.

Section 5.1 investigates question (a) on UCI data sets. Question (b) is investigated in Section 5.1 and 5.2 on UCI and real-world data sets, respectively. Question (c) is studied in Section 5.3. In the experiments, three state-of-the-art ALC algorithms, namely PMActive [Wu *et al.*, 2013], IEThresh [Donmez *et al.*, 2009] and ALC [Yan *et al.*, 2011], were evaluated as compared methods. In every experiment, each algorithm was repeated 20 times. If not specified explicitly, the linear kernel was employed in the classifier ($K(\mathbf{x}, \mathbf{y}) = \mathbf{x} \cdot \mathbf{y}$) and the RBF kernel was in reliability models ($K_t(\mathbf{x}, \mathbf{y}) = exp(-||\mathbf{x} - \mathbf{y}||^2)$).

5.1 Experiments on Simulated Data

To answer question (a), the three existing algorithms were applied in two scenarios, ALC scenario and ALCU scenario. In this experiment, each annotator is assigned with an area of expertise. For the instances belonging to its expertise, annotator t will give the correct labels with probability P_t and wrong labels with probability 1-P_t in both the ALC and ALCU scenarios. For the other instances, annotator t will give correct labels with probability p_t and wrong labels with 1-p_t in the ALC scenario while give unsure feedbacks in the ALCU scenario. The three compared methods mentioned above were developed for the ALC scenario. At the same time, they can be directly applied to the ALCU scenario by treating the unsure feedbacks as missing values (i.e., no responses). Although exactly the same algorithms were used in the two scenarios, we denote the algorithms used in the latter as PMActive+R, IETresh+R and ALC+R to differentiate the two scenarios. It is noteworthy that none of the three methods was modified to exploit the information contained in an unsure response, i.e., they just discarded the unsure responses. Hence, comparing the results of the three methods in the ALCU scenario against their results in the ALC scenario would allow us to assess whether it would be beneficial to provide annotators the unsure option.

The ALCU-SVM was also applied in the ALCU scenario and was compared against PMActive+R, IETresh+R and ALC+R. As ALCU-SVM explicitly makes use of the unsure responses to model the reliability of annotators, such a comparison would provide some evidence on whether it is worth developing a specially tailored approach, e.g., ALCU-SVM, for ALCU rather than utilizing some existing ALC directly.

When simulating multiple annotators with different expertise, we followed the settings in [Yan *et al.*, 2011] and [Wu *et al.*, 2013]: Each data set was first clustered into five subsets using k-means [Jain *et al.*, 1999]. This procedure was repeated twice and ten clusters were obtained. Then, 10 annotators were assumed such that annotator t ($t = 1, 2, \ldots, 10$) was capable of giving the correct labels for instances in cluster t with probability P_t, i.e., s/he would give the wrong label with probability $1 - P_t$. For the instances not belonging to cluster t, annotator t did not have enough knowledge. It was assumed to reply with unsure feedbacks in the ALCU scenario. In the ALC scenario, it would give the correct labels with probability p_t and make a mistake with $1 - p_t$. In our experiment, $P_t \sim U(0.9, 1.0)$ and $p_t \sim U(0.5, 0.6)$. Besides, each data set was randomly divided into three parts: initial set, active learning set and testing set. To be specific, the three data sets were divided as: Pima (20,548,200); Heart (20,150,100) and Ionosphere (20,180,151), where the three elements in the parenthesis are the numbers of instances in initial set, active learning set, and testing set respectively. As the same conclusion can be drawn for all the three data sets, only the results obtained on Pima data set are presented and analyzed below. The results on the other two data sets are available online[1].

It is deemed that wrong labels are harmful to active learning. The average numbers of wrong labels received by the 3 existing algorithms in ALC and ALCU scenarios are depicted in Fig. 1(a). It can be seen that all the methods in the ALCU scenario indeed have far fewer wrong labels, which demonstrates the benefit to provide annotators the unsure option.

[1]http://staff.ustc.edu.cn/ ketang/codes/IJCAI15ALCU.html

Fig. 1(b) plots the average accuracies of all the competing methods. It can be observed that ALCU-SVM outperformed the other methods in terms of the accuracy of the obtained classifier. This implies that the active learning procedure can be enhanced if the unsure feedback is exploited properly. In addition, querying labels of instances usually induces some sort of cost. Thus, an unsure feedback is in general unfavorable. In Fig. 1(c), the average cumulative numbers of unsure feedbacks received by ALCU-SVM and the other 3 methods in ALCU scenario are plotted. It can be observed that, by properly exploiting the unsure feedbacks, ALCU-SVM also managed to induce much fewer unsure feedbacks than the compared algorithms.

Suppose ALCU-SVM is run m iterations and there are N unlabeled instances in the beginning. In i-th iteration, the simplest way to optimize Eq. (9) is to calculate the value for all $N - i + 1$ unlabeled instances at first and then choose the instance which minimizes Eq. (9). Moreover, getting the value of $\Omega(\mathbf{x})$ involves T logical operations (see Eq. (3)). Hence, the complexity of ALCU-SVM is $\sum_{i=1}^{m}(N\text{-}i\text{+}1)T$, i.e. $O(mNT)$. Fig. 2 plots the average running time of all the seven competing methods on the three data sets. It can be seen that ALCU-SVM is one of the least time-consuming methods. Comparing all the methods in ALC scenario with the corresponding methods in the ALCU scenario, the methods in the latter situation are in general less costly. For example, ALC+R needs much less time than ALC. One possible reason may be that the number of noise labels which slow down the convergence of algorithm is largely decreased by the unsure option.

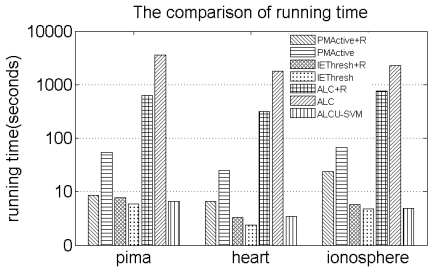


Figure 2: The comparison of average running time

5.2 Experiments on Real Data

In addition to preliminary empirical studies on UCI data, experiments have also been carried out in a real-world ALCU scenario. In this experiment, the UDI-TwitterCrawl-Aug2012-Tweets[2] [Li *et al.*, 2012], which include 50 million tweets posted mainly between 2008 and 2011, were employed. Concretely, the hashtags of tweets were used as their "labels". The tweets with such hashtags as job, music, royal wedding (Prince William and Kate Middleton) or Osama Bin Laden being killed were used in our experiment. In total 1000 tweets belonging to these 4 topics were randomly chosen as the training data, and another 1000 tweets were chosen as the testing data. The text body of each tweet was pre-processed and transferred into a TF-IDF vector by the natural language toolkit (NLTK[3]), and it ended uabelson-et-al:schemeabelson-et-al:schemep with a numerical feature matrix with 295 features. As binary-class problems are considered in this work, the tweets belonging to the "job" topic were treated as positive class, and the tweets under the other 3 topics were treated as negative class.

Given the above-described training and testing data, 5 real annotators (i.e., human) from our university were invited to assist in our experiment. Specifically, all of them were first asked to label 200 training tweets for initial training. Then, labels of the other 800 instances were queried from the annotators during the active learning procedure. For each tweet, an annotator could either label it as one of the 4 topics or reply "I'm unsure" to the query.

The experiment was conducted in a real situation with an unsure option, i.e., PMActive+R, IETresh+R, ALC+R and ALCU-SVM were compared. The average accuracies achieved by the competing methods are plotted in Fig. 3(a). It can be seen that ALCU-SVM dominates all of the other methods. Considering that all the competing methods ran in the same situation with the unsure option, it verifies that it is worth investigating a specially tailored approach for ALCU rather than applying some existing ALC directly. The average numbers of unsure feedbacks are plotted in Fig. 3(b), which also shows the advantage of ALCU-SVM in this aspect. The advantages of ALCU-SVM observed in Fig. 3(b) is not as significant as that observed in Fig. 1(b). One possible reason is that the ratio of unsure feedback in this experiment is much smaller than that in the experiments on UCI datasets. For example, it can be seen later in Table 1, annotators 2, 3 and 5 replied with unsure feedbacks in about 200 out of 1000 instances, while about 4/5 of replies were unsure feedbacks in the experiments on UCI datasets.

	Correct labels	Wrong labels	Unsure feedbacks
A_1	580	22	398
A_2	732	33	235
A_3	660	73	267
A_4	425	89	486
A_5	741	47	212

Table 1: The labeling result of each annotator

	A_1	A_2	A_3	A_4	A_5
A_1	1	0.76	0.724	0.627	0.718
A_2	0.76	1	0.796	0.593	0.827
A_3	0.724	0.796	1	0.595	0.772
A_4	0.627	0.593	0.595	1	0.588
A_5	0.718	0.827	0.772	0.588	1

Table 2: The similarities between annotators

To further assess the necessity of ALCU in real-world scenarios, the 5 annotators were also requested to provide labels for all the 1000 training instances. The numbers of instances that each annotator provided correct/wrong labels and unsure feedbacks are summarized in Table 1, which shows that all annotators are unsure about the labels of a substantial number of instances, indicating that without an unsure option, a large number of poor quality labels could be introduced. Moreover,

[2]https://wiki.cites.illinois.edu/wiki/display/forward/Dataset-UDI-TwitterCrawl-Aug2012

[3]http://www.nltk.org

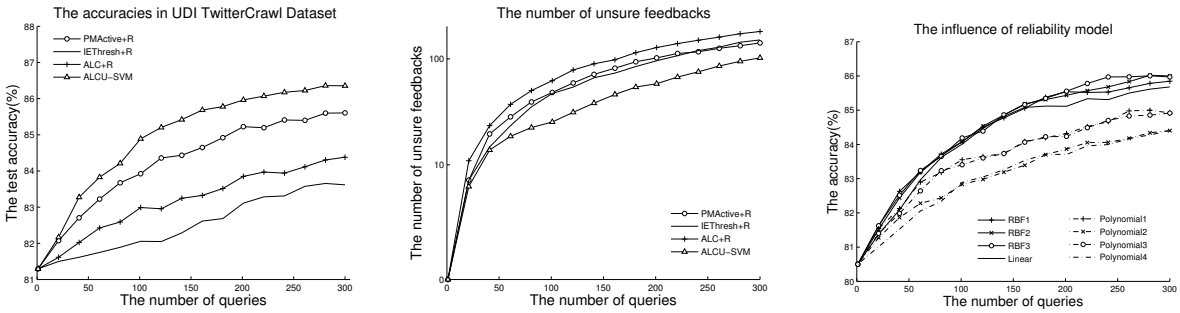


Figure 3: The results on real dataset.

Table 2 presents the pair-wise similarities between annotators' labeling behaviors. Each element in the table represents the ratio of instances in which the two corresponding annotators provided the same labels. The observed small similarity values indicate that the 5 human annotators involved in the experiment behaved rather differently. This observation justified that for a real-world problem, it would be important to model the reliability of each annotator since each may show diverse labeling behaviors.

5.3 Influence of Reliability Model

A few hyper-parameters, i.e., the kernel function and its parameters, need to be predefined for ALCU-SVM to learn the reliability models. In previous experiments, only the RBF kernel was employed. To assess the influence of this issue on ALCU-SVM, further experiments have been conducted on the real-world data set using 8 different settings of kernel functions and hyper-parameters:

1. Linear: $\mathbf{x} \cdot \mathbf{y}$
2. RBF1: $e^{-||\mathbf{x}-\mathbf{y}||^2}$
3. RBF2: $e^{-5||\mathbf{x}-\mathbf{y}||^2}$
4. RBF3: $e^{-10||\mathbf{x}-\mathbf{y}||^2}$
5. Polynomial1: $(\mathbf{x} \cdot \mathbf{y})^2$
6. Polynomial2: $(\mathbf{x} \cdot \mathbf{y})^{10}$
7. Polynomial3: $(2\mathbf{x} \cdot \mathbf{y})^2$
8. Polynomial4: $(2\mathbf{x} \cdot \mathbf{y})^{10}$

Fig. 3(c) plots the results of all these 8 kernels. It can be observed that both the kernel type and hyper-parameters affect the accuracy of ALCU-SVM. However, the kernel type appears to have much more significant influence than the hyper-parameters. Specifically, the polynomial kernel led to significantly lower performance than the other two kernels, while the RBF and linear kernel led to comparable accuracy. Therefore, a rule-of-thumb in practice would be to first employ both the linear and RBF kernels for a few iterations and then test the obtained reliability model against newly obtained responses from annotators. After that, the kernel that better fit the distribution of the annotators' expertise can be chosen.

6 Conclusion and Discussion

In this paper, a variant of active learning from crowds, namely active learning from crowds with unsure option, is put forward and investigated. An algorithm called ALCU-SVM is proposed for this new problem. Empirical studies on both simulated data and real-world crowdsourcing data imply that providing annotators the unsure option would benefit ALC significantly. Further, although some existing ALC methods are directly applicable to ALCU, they were clearly outperformed by the proposed ALCU-SVM in the empirical studies. This suggests that exploiting the unsure feedbacks to model the expertise (e.g., in forms of reliability) of annotators, as ALCU-SVM does, is crucial for tackling ALCU. Hence, it is worth investigating specially tailored approaches to this new learning problem.

There are a few directions for further improving ALCU-SVM. First, ALCU-SVM solely relies on the feedbacks of annotators to decide whether an annotator will be trusted in a given instance. This setting may suffer in case annotators are falsely confident. A possible solution would be to employ two models to represent annotators' expertise. That is, the reliability model used in ALCU-SVM, which reflects the confidence of annotators, and a probabilistic model that represents the probability that an annotator can correctly label an instance. The latter can be used for filtering possibly wrong labels provided by a falsely confident annotator. On the other hand, making ALCU-SVM more robust to wrong labels would be another direction to resolve this problem. Second, ALCU-SVM takes a binary view of the annotators' reliability, e.g., Ω is defined as the disjunction over all annotators. Other forms of functions can be employed for Ω so as to avoid the instance selection being biased to instances in which at least one annotator is deemed to be sufficiently reliable. Potential approaches could be using a soft-max function for Ω or to seek more advanced techniques, e.g., [Auer *et al.*, 2002] [Liu *et al.*, 2009], to guarantee that a reliable annotator will be found if any exists. Third, theoretical properties of ALCU-SVM are also worthy of further investigations to fully understand the advantages and disadvantages of ALCU-SVM.

In additional to improving ALCU-SVM, it would also be interesting (and maybe even more important) to explore alternative frameworks to tackle ALCU. A thread for thoughts is to view the problem from the perspective of optimal annotator selection, and the family of Dynamic Classifier Selection (DCS) methods [Ho *et al.*, 1994] [Woods *et al.*, 1996], could be used. Annotator selection may also be explicitly formulated as a bandit problem, where the reward of choosing an

annotator is high/low if the annotator gives a label/says "unsure". In this sense, some bandit algorithms [Gittins *et al.*, 2011] might be applicable to ALCU after appropriate modifications. Last but not least, if it is possible to recruit a large number of human annotators, it will be very interesting to study the scalability of the ALCU-SVM (and any other new algorithm proposed for ALCU) with respect to number of annotators on a real crowdsourcing platform.

7 Acknowledgement

This work was supported in part by the 973 Program of China under Grants 2011CB707006 and 2014CB340501, the National Natural Science Foundation of China under Grants 61175065, 61329302, 61333014 and 61321491, the Program for New Century Excellent Talents in University under Grant NCET-12-0512, and the European Union Seventh Framework Programme under Grant 247619.

References

[Auer *et al.*, 2002] P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002.

[Batuwita and Palade, 2010] R. Batuwita and V. Palade. Fsvm-cil: fuzzy support vector machines for class imbalance learning. *IEEE Transactions on Fuzzy Systems*, 18(3):558–571, 2010.

[Dagan and Engelson, 1995] I. Dagan and S. P. Engelson. Committee-based sampling for training probabilistic classifiers. In *ICML*, pages 150–157, 1995.

[Dekel *et al.*, 2012] O. Dekel, C. Gentile, and K. Sridharan. Selective sampling and active learning from single and multiple teachers. *JMLR*, 13(1):2655–2697, 2012.

[Donmez *et al.*, 2009] P. Donmez, J. G. Carbonell, and J. Schneider. Efficiently learning the accuracy of labeling sources for selective sampling. In *KDD*, pages 259–268. ACM, 2009.

[Friedel *et al.*, 2006] C. C. Friedel, U. Rückert, and S. Kramer. Cost curves for abstaining classifiers. In *ICML 2006 Workshop on ROC Analysis*, pages 33–40. Citeseer, 2006.

[Gittins *et al.*, 2011] J. Gittins, K. Glazebrook, and R. Weber. *Multi-armed bandit allocation indices*. John Wiley & Sons, 2011.

[He and Garcia, 2009] H. He and E. A. Garcia. Learning from imbalanced data. *TKDE*, 21(9):1263–1284, 2009.

[Ho *et al.*, 1994] T. K. Ho, J. J. Hull, and S. N. Srihari. Decision combination in multiple classifier systems. *TPAMI*, 16(1):66–75, 1994.

[Huang *et al.*, 2010] S.-J. Huang, R. Jin, and Z.-H. Zhou. Active learning by querying informative and representative examples. In *NIPS*, pages 892–900, 2010.

[Jain *et al.*, 1999] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM computing surveys*, 31(3):264–323, 1999.

[Kwok, 1999] J.-Y. Kwok. Moderating the outputs of support vector machine classifiers. *IEEE Transactions on Neural Networks*, 10(5):1018–1031, 1999.

[Lewis and Catlett, 1994] D. D. Lewis and J. Catlett. Heterogeneous uncertainty sampling for supervised learning. In *ICML*, volume 94, pages 148–156, 1994.

[Li *et al.*, 2012] R. Li, S. Wang, H. Deng, R. Wang, and K. C.-C. Chang. Towards social user profiling: unified and discriminative influence model for inferring home locations. In *KDD*, pages 1023–1031. ACM, 2012.

[Liu *et al.*, 2009] X.-Y. Liu, J. Wu, and Z.-H. Zhou. Exploratory undersampling for class-imbalance learning. *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, 39(2):539–550, 2009.

[Long *et al.*, 2013] C. Long, G. Hua, and A. Kapoor. Active visual recognition with expertise estimation in crowdsourcing. In *ICCV*, pages 3000–3007. IEEE, 2013.

[Nguyen and Smeulders, 2004] H. T. Nguyen and A. Smeulders. Active learning using pre-clustering. In *ICML*, page 79. ACM, 2004.

[Pietraszek, 2005] T. Pietraszek. Optimizing abstaining classifiers using roc analysis. In *ICML*, pages 665–672. ACM, 2005.

[Roy and McCallum, 2001] N. Roy and A. McCallum. Toward optimal active learning through monte carlo estimation of error reduction. *ICML*, 2001.

[Settles *et al.*, 2008] B. Settles, M. Craven, and S. Ray. Multiple-instance active learning. In *NIPS*, pages 1289–1296, 2008.

[Settles, 2010] B. Settles. Active learning literature survey. *University of Wisconsin, Madison*, 52:55–66, 2010.

[Sheng *et al.*, 2008] V. S. Sheng, F. Provost, and P. G. Ipeirotis. Get another label? improving data quality and data mining using multiple, noisy labelers. In *KDD*, pages 614–622. ACM, 2008.

[Woods *et al.*, 1996] K. Woods, K. Bowyer, and Kegelmeyer W. P. Combination of multiple classifiers using local accuracy estimates. In *CVPR*, pages 391–396. IEEE, 1996.

[Wu *et al.*, 2013] W. Wu, Y. Liu, M. Guo, C. Wang, and X. Liu. A probabilistic model of active learning with multiple noisy oracles. *Neurocomputing*, 118:253–262, 2013.

[Yan *et al.*, 2011] Y. Yan, G. M. Fung, R. Rosales, and J. G. Dy. Active learning from crowds. In *ICML*, pages 1161–1168, 2011.

[Yan *et al.*, 2012] Y. Yan, R. Rosales, G. Fung, F. Farooq, B. Rao, and J. G. Dy. Active learning from multiple knowledge sources. In *ICAIS*, pages 1350–1357, 2012.

[Zhao *et al.*, 2014] L. Zhao, Y. Zhang, and G. Sukthankar. An active learning approach for jointly estimating worker performance and annotation reliability with crowdsourced data. *arXiv preprint*, 2014.

[Zhou, 2012] Z.-H. Zhou. *Ensemble methods: foundations and algorithms*. CRC Press, 2012.