Efficient Crowd-Powered Active Learning for Reliable Review Evaluation

Xinping Min*, Yuliang Shi*, Lizhen Cui*, Han Yu‡ and Chunyan Miao†‡
*School of Computer Science and Technology, Shandong University, China
†School of Computer Science and Engineering (SCSE), Nanyang Technological University (NTU), Singapore
‡Joint NTU-UBC Research Centre of Excellence in Active Living for the Elderly (LILY), NTU, Singapore
minxinping0105@126.com, {shiyuliang, clz, lqz}@sdu.edu.cn, {han.yu, ascymiao}@ntu.edu.sg

***Abstract*—To mitigate uncertainty in the quality of online purchases (e.g., e-commerce), many people rely on review comments from others in their decision-making processes. The key challenge in this situation is how to identify useful comments among a large corpus of candidate review comments with potentially varying usefulness. In this paper, we propose the Reliable Review Evaluation Framework (RREF) which combines crowdsourcing with machine learning to address this problem. To improve crowdsourcing quality control, we propose a novel review query crowdsourcing approach which jointly considers workers' track records in review provision and current workloads when allocating review comments for workers to rate. Using the ratings crowdsourced from workers, RREF then enhances the adaptive topic classification model selection and weighting functions of AdaBoost with dynamic keyword list reconstruction. RREF has been compared with state-of-the-art related frameworks using a large-scale real-world dataset, and demonstrated over 50% reduction in average classification errors.**

I. Introduction

Electronic services, such as e-commerce, are often provided by people with diverse skills, resources or motives. There is significant uncertainty, especially when a user consumes such a service. There is a strong need from users for review comments from previous customers in order to reduce the uncertainty in their decision processes. This has given rise to many review service providers such as *TripAdvisor* where users can post comments about a wide variety of such services for others to browse.

The success of these review service providers has created another problem - the proliferation of review comments. For example, in 2015, TripAdvisor users posted over 320 million reviews.[1] This makes it challenging for a user to manually look through the typically large number of reviews associated with an entity to identify those reviews that are most useful. Thus, review recommendation services powered by natural language processing (NLP) techniques can be a useful tool to help users solve this problem. Training effective NLP models typically requires well-labelled data. However, employing professional annotators to produce the labels is often costly and time consuming. In recent years, crowdsourcing has become an increasingly popular approach for acquiring labels for large training datasets [1].

Crowdsourcing refers to the process of harnessing the contributions of a large number of people, typically through the Internet, to complete tasks [2]. Successful examples of crowdsourcing include the reCAPTCHA system which provides online security while utilizing the crowd's efforts to digitize pre-computer era books [3], and fold.it which allows the crowd to contribute towards scientific discoveries [4]. Existing literature in Active Learning has realized the advantages of dynamically obtaining labels via crowdsourcing [5], [6], and accounting for the workers' abilities [7], [8]. Thus, it is important to provide quality control mechanisms when crowdsourced labels are to be used to train machine learning models.

In this paper, we propose the Reliable Review Evaluation Framework (RREF) which infuses crowd intelligence into active machine learning. It contains two novel approaches to enhance the accuracy of identifying useful review comments to be recommended to users. Firstly, a situation-aware task allocation approach is proposed which performs crowdsourcing quality control. It jointly considers each worker's track records in review rating (i.e., label) provision and current workload to make optimal trade-offs between the overall label quality and time elapsed. Secondly, RREF enhances the AdaBoost framework by combining dynamic keyword list reconstruction with adaptive selection and re-weighting of multiple topic classification models to improve overall classification accuracy.

We prove that the task allocation strategies computed by RREF can achieve close to the optimal overall accuracy and that the proximity to the optimal value can be controlled easily through trade-offs in waiting time. Furthermore, RREF has been extensively evaluated using a large-scale review comments dataset crawled from Amazon China. Compared to two state-of-the-art frameworks, RREF achieves more than 50% reduction in average classification error rates.

II. Related Work

There is considerable work in the literature on NLP through traditional machine learning approaches [9], [10], [11], [12]. Techniques such as Latent Dirichlet Allocation (LDA) [13] are often used.

As well-labelled training data are important for machine learning, researchers are always looking for ways to gather good quality labels at scale. In [1], [14], the authors have shown that a multitude of workers can potentially produce

[1] http://www.tripadvisor.com/PressCenter-c4-Fact_Sheet.html

useful labelled data for NLP tasks as well as experts can. Following these successes in utilizing crowd intelligence, recently, active learning frameworks which include human labellers into the machine learning process through crowdsourcing are starting to emerge. In [15], the authors proposed a framework which adopts crowdsourced labels from workers and experts to reduce the noise in the labels.

The framework proposed in [16] - JCF - is the most closely related to our proposed RREF in this paper. JCF combines crowdsourced review comment ratings with active learning SVM classifiers with the aim to improve classification accuracy. Expert supervisors' opinions are used to adjust the Random Active Learning SVM and the Margin-based Active Learning SVM, while opinions from common crowdsourcing workers are used to adjust the Crowd Active Learning SVM classifier.

Nevertheless, existing frameworks do not possess quality control mechanism for the crowdsourcing step. Nor do they adjust the keyword vectors during the active learning process in response to the crowd's feedback. In the following section, we explain how the proposed RREF addresses these limitations.

III. The Proposed RREF

Our goal is to construct a crowdsourcing-powered machine learning framework which can accurately identify useful review comments as shown in Figure 1. Based on the rating information (i.e., the labels) provided by crowdsourcing workers, we aim to improve the classification model accuracy.

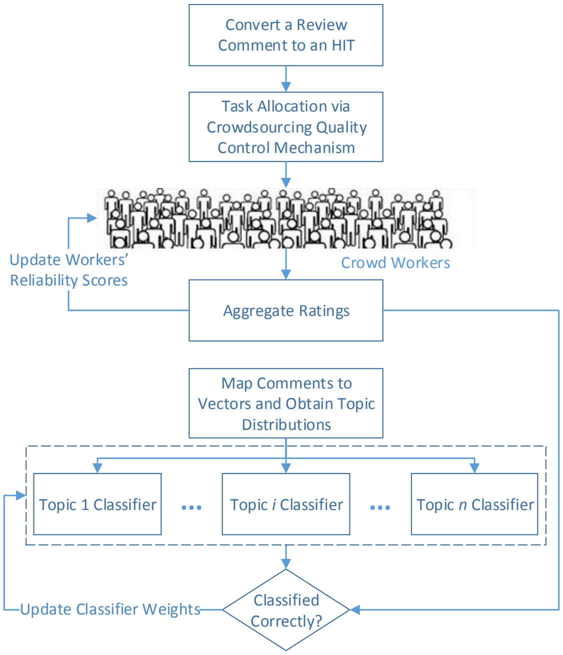
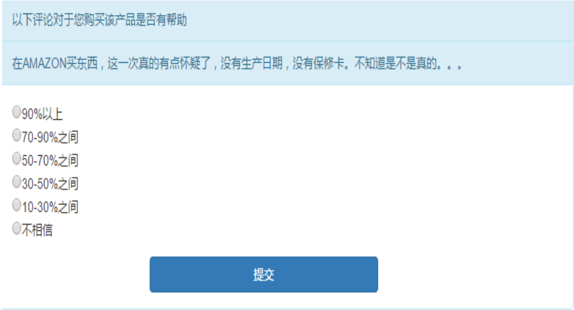

Convert a Review Comment to an HIT
Task Allocation via Crowdsourcing Quality Control Mechanism
Update Workers' Reliability Scores
Crowd Workers
Aggregate Ratings
Map Comments to Vectors and Obtain Topic Distributions
Topic 1 Classifier
...
Topic *i* Classifier
...
Topic *n* Classifier
Update Classifier Weights
Classified Correctly?

Fig. 1. The system architecture of RREF.

When new review comments are received, the proposed framework first maps these comments into human intelligence tasks (HITs) using our HIT template. Then, these HITs are dynamically allocated to a multitude of workers to obtain their ratings. The quality and timeliness of the crowdsourced review comment ratings are optimized by the proposed HIT allocation approach.

While workers are working on the HITs, the framework calculates distributions of different topics and maps the comments into vectors. The comment classification model will then compute the usefulness scores for the comments. When ratings from the workers are received, they are used to update the weight values assigned to each classifier so that the workers' opinions can be used to guide the adjustment of relative importance given to each classifier. Once the weight values converge, the training process terminates and a classification model with accuracies improved by human intelligence can be obtained.

A. *Crowdsourcing Quality Control*



以下评论对于您购买该产品是否有帮助

在AMAZON买东西，这一次真的有点怀疑了，没有生产日期，没有保修卡。不知道是不是真的。。。

○90%以上
○70-90%之间
○50-70%之间
○30-50%之间
○10-30%之间
○不相信

Fig. 2. Our rating request HIT template.

Review comments, especially those written in Chinese language, may contain subtleties which are difficult for today's natural language processing techniques to recognize. However, they can be very clear to a speaker of the language. Thus, it is advantageous to integrate ratings about the usefulness of review comments from humans through crowdsourcing into the machine learning process. For this purpose, we designed a crowdsourcing HIT template as shown in Figure 2 which displays a review comment to a worker and allows him to select how useful he feels the comment can be to his decision-making process.

Nevertheless, when obtaining such ratings, the heterogeneity in workers' experience and reliability will inevitability affect the quality of their comments. Thus, it is not advantageous to simply rely on the traditional mode of crowdsourcing where the request for comments about a service is just advertised to workers waiting for them to respond (similar to Amazon's Mechanical Turk (mTurk)[2]).

To address this problem, we propose a dynamic rating crowdsourcing approach. Whenever new review comments arrive, they are automatically composed into HITs by RREF following our HIT template. With tracking tools such as *Turkalytics* [17], workers' behaviour related data in a given crowdsourcing system can be monitored efficiently. Then, the

[2] https://www.mturk.com/mturk/welcome

HITs are allocated to a multitude of workers based on the proposed crowdsourcing approach to obtain ratings.

Let $Q_w(t)$ be the number of rating requests in a worker w's pending task queue at time slot t. The queueing dynamics can be expressed as:

$$Q_w(t+1) = \max[0, Q_w(t) + a_w(t) - c_w(t)] \tag{1}$$

where $a_w(t)$ is the number of new rating requests sent to worker w at time slot t, and $c_w(t)$ is the number of rating requests completed by worker w at time slot t.

A worker w's reliability at time slot t based on his track records is denoted by $\gamma_w(t) \in [0, 1]$. In this paper, we adopt the model proposed in [18] to compute the value of $\gamma_w(t)$. The model computes not only worker's probability of producing a useful rating, but also the uncertainty based on the amount of available track records. Thus, $\gamma_w(t)$ represents the probability of a worker w's rating for a comment being useful for improving our machine learning model discounted by the uncertainty involved. Nonetheless, other models for computing a worker's reliability [19] can also be used in conjunction with the proposed crowdsourcing approach as long as the outputs can be normalized to the range of $[0, 1]$.

Let $U(t)$ be the overall utility (i.e., the expected accuracy of ratings obtained) of allocating requests for ratings to a given crowd of N workers at time slot t. We have:

$$U(t) = \sum_{w=1}^{N} \gamma_w(t) a_w(t). \tag{2}$$

Recent findings in social science suggests that human choice behaviour can be accounted for by a mixture of *utility maximization* and *surprise minimization* [20]. We adopt this principle in our crowdsourcing approach in order to improve the chance of the computed request allocation plan to be accepted by the workers. To this end, we need a way to model the concept of *surprise* for workers working on the requests. The variation in workload for a worker can be regarded as a form of surprise, which shall be minimized.

The *Lyapunov function* [21], which measures the overall congestion of demand on workers at time slot t, can be expressed as:

$$F_L(t) = \frac{1}{2} \sum_{w=1}^{N} Q_w^2(t) \tag{3}$$

Let $\mathbf{Q}(t)$ denote a vector of all workers' pending task queues at time slot t. Using the *Lyapunov drift* $\Delta(\mathbf{Q}(t))$ as a measure of the variations in workers' workloads, we have:

$$\begin{aligned}
\Delta(\mathbf{Q}(t)) &= \mathbb{E}\{F_L(t+1) - F_L(t)|\mathbf{Q}(t)\} \\
&= \frac{1}{T} \sum_{t=0}^{T-1} \sum_{w=1}^{N} \left(\frac{1}{2} Q_w^2(t+1) - \frac{1}{2} Q_w^2(t) \right) \\
&= \frac{1}{T} \sum_{t=0}^{T-1} \sum_{w=1}^{N} \Bigg(Q_w(t) a_w(t) \\
&\quad - c_w(t)[Q_w(t) + a_w(t)] + \frac{1}{2}[a_w^2(t) + c_w^2(t)] \Bigg) \\
&\leqslant \frac{1}{T} \sum_{t=0}^{T-1} \sum_{w=1}^{N} [Q_w(t) a_w(t) + \Psi]
\end{aligned} \tag{4}$$

where $\Psi = \frac{1}{2}[a_{\max}^2 + c_{\max}^2] \geqslant 0$. $a_{\max}$ and $c_{\max}$ are the physical upper limits on $a_w(t)$ and $c_w(t)$ for all w and t in a given crowdsourcing system, and can be regarded as constants for our purpose.

Thus, we can formulate the *utility-minus-surprise* objective function as:

$$\begin{aligned}
&\sigma \mathbb{E}\{U(t)|\mathbf{Q}(t)\} - \Delta(\mathbf{Q}(t)) \\
&\geqslant \frac{1}{T} \sum_{t=0}^{T-1} \sum_{w=1}^{N} (\sigma \gamma_w(t) a_w(t) - Q_w(t) a_w(t) - \Psi)
\end{aligned} \tag{5}$$

where $\sigma > 0$ is a weight factor determining the relative importance given to maximizing utility versus minimizing surprise. By considering only the terms containing the solution variables $a_w(t)$ for all workers, the optimization can be formulated as:

Maximize:

$$\frac{1}{T} \sum_{t=0}^{T-1} \sum_{w=1}^{N} a_w(t)[\sigma \gamma_w(t) - Q_w(t)] \tag{6}$$

Subject to:

$$\gamma_w(t) \geqslant \gamma_{\min}, \forall w, t \tag{7}$$

$$a_w(t) \leqslant c_{\max}, \forall w, t \tag{8}$$

where $\gamma_{\min}$ is the minimum reliability value a worker w must achieve in order to be allowed to participate. The value can be set by the system administrator. By setting the upper limit of $a_w(t)$ to $c_{\max}$ in Constraint 8, we intend to keep the workers busy. Nevertheless, other strategies can be easily implemented by changing Constraint 8.

The solution variables for the above optimization are $a_w(t) \in \{0, \mathbb{Z}^+\}$ for all workers whenever there are new review comments to be rated by multiple workers. To maximize Eq. (6), at each time slot t, Algorithm 1 computes the values of the expression $[\sigma \gamma_w(t) - Q_w(t)]$ (denoted as $\eta_w(t)$ for simplicity) for every worker w. It then sorts all workers in descending order of their $\eta_w(t)$ values. For each worker w who satisfies Constraint (7), set $a_w(t)$ based on Constraints (8). The rating request allocation terminates when there are no more workers with $\eta_w(t) > 0$.

Once the ratings for a comment are received from all workers, they are aggregated through majority voting. Then, the

Algorithm 1 Rating Request Allocation

```
Require: Number of review comments $N_c(t)$ which need to be rated at a given time $t$; $\sigma$; $Q_w(t)$, $c_{\max}$ and $\gamma_w(t)$ values for all workers.
Compute $\eta_w(t)$ for all $w$;
Rank all $w$ in descending order of $\eta_w(t)$;
for each worker $w$ do
    if $\eta_w(t) > 0$ and $\gamma_w(t) \geqslant \gamma_{\min}$ then
        if $N_c(t) < c_{\max}$ then
            $a_w(t) = N_c(t)$;
        else
            $a_w(t) = c_{\max}$;
        end if
    else
        $a_w(t) = 0$;
    end if
end for
return $\{a_1(t), a_2(t), ...\}, \forall w$;
```

workers' performance for this task is evaluated by comparing their individual ratings against the aggregated rating. Workers whose ratings are the same as the aggregated rating will have their reliability scores increased, while those whose ratings differ from the aggregated rating will have their reliability score reduced following [18].

B. Crowd-Powered Adaptive Classifier Selection

In order to make the proposed framework flexible in adapting what classification technique to employ with changing situation, we adopt AdaBoost [22] as the basis for RREF. AdaBoost provides an effective framework for combining multiple classifiers to enhance the overall performance. It consists of two steps. Firstly, suppose we have selected m classifiers and need to select an additional classifier from the rest of available classifiers, the one with the minimum misclassification cost among all remaining classifiers shall be selected. Secondly, the weight for the newly added classifier is determined by minimizing the total misclassification cost.

With the help from human labellers through crowdsourcing, the weight of each training sample can be obtained during the training process. The training sample weights are updated according to the following principle. Misclassified samples are assigned higher weight values while the weight values for correctly classified samples are reduced. In this way, RREF can focus on selecting a new classifier which can play a more significant role in improving the classification accuracy for the misclassified samples, thereby improving the overall performance.

The classification result of a model containing M classifiers can be expressed as a linear combination:

$$\begin{aligned} C_M(\mathbf{T_i}) &= \sum_{m=1}^{M} \omega_m k_m(\mathbf{T_i}) \\ &= C_{(M-1)}(\mathbf{T_i}) + \omega_M k_M(\mathbf{T_i}) \end{aligned} \tag{9}$$

where $\omega_m > 0$ represents the weight of the m-th classifier. Classifiers with higher error rates are given lower weights. $k_m(\mathbf{T_i})$ represents the classification output of a classifier k_m on a multidimensional input feature vector $\mathbf{T_i}$ (i.e., a keyword list representing a review comment).

Suppose there are already $M-1$ classifiers included in the AdaBoost model, and we want to select one more classifier from a pool of available classifiers to join the cascade.

The problem becomes how to select a classifier from the classifier pool given a training set $\{\{\mathbf{T_1}, u_1\}, \{\mathbf{T_2}, u_2\}, ..., \{\mathbf{T_n}, u_n\}\}$ in which a comment $\mathbf{T_i}$ has a usefulness score u_i. To address this problem, the error function ϵ can be expressed as follows:

$$\epsilon = \sum_{i=1}^{n} \alpha_i^m e^{-\omega_m f(u_i, k_m(\mathbf{T_i}))} \tag{10}$$

where

$$\alpha_i^m = \begin{cases} 1 & , \text{ if } m = 1 \\ e^{-u_i C_{(m-1)}(\mathbf{T_i})} & , \text{ otherwise.} \end{cases} \tag{11}$$

The function $f(u_i, k_m(\mathbf{T_i}))$ is defined as:

$$f(u_i, k_m(\mathbf{T_i})) = \begin{cases} -1, & \text{if } k_m(\mathbf{T_i}) \neq u_i \\ 1, & \text{otherwise.} \end{cases} \tag{12}$$

Thus, ϵ can be re-written as:

$$\epsilon = \sum_{k_m(\mathbf{T_i}) \neq u_i} \alpha_i^m e^{\omega_m} + \sum_{k_m(\mathbf{T_i}) = u_i} \alpha_i^m e^{-\omega_m} \tag{13}$$

To determine the updated weight values ω_m which minimize the overall error rate ϵ, we find the first order derivative of ϵ with respect to ω_m and equate it to 0. Solving the differential

Algorithm 2 keyword list reconstruction

```
Require: A keywords list $\mathbf{K}$; a review comment $\mathbf{T_i}$ containing a set of keywords.
Vector_ItemList = $\{\emptyset\}$;
if Predicted label for a comment $\mathbf{T_i}$ differs significantly from the crowd annotated label for $\mathbf{T_i}$ then
    Rank all keywords in $\mathbf{K}$ in ascending order of their contributions to classification accuracy;
    Select the top $k$ keywords from the re-ordered $\mathbf{K}$ to form a replacement list $\mathbf{I_r}$;
    Randomly select keywords from $(\mathbf{K} - \mathbf{I_r})$ to replace the keywords in $\mathbf{I_r}$;
    Combine the replaced $\mathbf{I_r}$ with $(\mathbf{K} - \mathbf{I_r})$ to form Vector_ItemList;
end if
return Vector_ItemList;
```

equation yields:

$$\begin{aligned}\omega_m &= \frac{1}{2}\ln\left(\frac{\sum_{k_m(\mathbf{T_i})=u_i}\alpha_i^m}{\sum_{k_m(\mathbf{T_i})\neq u_i}\alpha_i^m}\right)\\ &= \frac{1}{2}\ln\left(\frac{\sum_{i=1}^{n}\alpha_i^m - \sum_{k_m(\mathbf{T_i})\neq u_i}\alpha_i^m}{\sum_{k_m(\mathbf{T_i})\neq u_i}\alpha_i^m}\right)\\ &= \frac{1}{2}\ln\left(\frac{1-\varepsilon}{\varepsilon}\right)\end{aligned} \tag{14}$$

where $\varepsilon = \frac{\sum_{k_m(\mathbf{T_i})\neq u_i}\alpha_i^m}{\sum_{i=1}^{n}\alpha_i^m}$ is the weighted error rate of a classifier m among all selected classifiers.

In practice, there can be cases in which one classifier predicts a result which differs too much from the ground truth. For instance, given a review comment $\mathbf{T_i}$, a classifier may label it as “over 90%” useful, while the aggregate label from crowdsourcing workers is “10% to 30% useful”. To deal with these situations, we further enhance AdaBoost by proposing the comment vector reconstruction method as shown in Algorithm 2.

IV. Analysis

In this section, we analyze the performance bounds of the proposed crowdsourcing quality control mechanism.

Assume there exist constants σ, ξ and φ at a given time slot t such that:

$$\sigma U(t) - \Delta(\mathbf{Q}(t)) \geqslant \sigma U^* + \xi \sum_{w=1}^{N} Q_w(t) - \varphi, \tag{15}$$

where U^* is the optimal rating accuracy produced by a mechanism which can perfectly predict the workers’ behaviour. We have:

$$\begin{aligned}&\sigma \sum_{w=1}^{N} \mathbb{E}\{\gamma_w(t)a_w(t)\}\\ &-\mathbb{E}\{F_L(Q_w(t+1)) - F_L(Q_w(t))\} \geqslant \sigma U^*\\ &+\xi \sum_{w=1}^{N} \mathbb{E}\{Q_w(t)\} - \varphi\end{aligned} \tag{16}$$

which holds for all time slots. By summing both sides of Eq. (16) over all $t \in \{0, 1, ..., T-1\}$, we have:

$$\begin{aligned}&\sigma \sum_{t=1}^{T-1}\sum_{w=1}^{N} \mathbb{E}\{\gamma_w(t)a_w(t)\}\\ &-\mathbb{E}\{F_L(Q_w(T)) - F_L(Q_w(0))\} \geqslant \sigma T U^*\\ &+\xi \sum_{t=1}^{T-1}\sum_{w=1}^{N} \mathbb{E}\{Q_w(t)\} - T\varphi.\end{aligned} \tag{17}$$

Since $Q_w(t) \geqslant 0$ for all w and t, $F_L(\cdot) \geqslant 0$ and $F_L(0) = 0$, Eq. (17) can be re-written as:

$$\begin{aligned}&\frac{1}{T}\sum_{t=1}^{T-1}\sum_{w=1}^{N} \mathbb{E}\{\gamma_w(t)a_w(t)\}\\ &\geqslant U^* + \frac{\xi}{\sigma T}\sum_{t=1}^{T-1}\sum_{w=1}^{N} \mathbb{E}\{Q_w(t)\} - \frac{\varphi}{\sigma}\\ &+\frac{1}{\sigma T}\mathbb{E}\{F_L(Q_w(T))\} \geqslant U^* - \frac{\varphi}{\sigma}.\end{aligned} \tag{18}$$

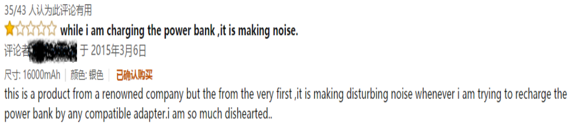
Therefore, we prove that the *lower bound* on the time averaged expected accuracy of the overall rating on review comments from workers achieved by following the proposed crowdsourcing quality control mechanism is within $\frac{\varphi}{\sigma}$ of that achieved by the theoretical optimal solution. If σ is increased to significantly large, most rating requests will concentrate on workers who have demonstrated good reliability in the past, thereby resulting in high rating accuracy.

Since $U^* \geqslant 0$, re-arranging the terms in Eq. (18) yields:

$$\begin{aligned}&\frac{1}{T}\sum_{t=1}^{T-1}\sum_{w=1}^{N} \mathbb{E}\{Q_w(t)\}\\ &\leqslant \frac{\sigma}{\xi T}\sum_{t=1}^{T-1}\sum_{w=1}^{N} \mathbb{E}\{\gamma_w(t)a_w(t)\}\\ &-\frac{\sigma}{\xi}U^* + \frac{\varphi}{\xi} - \frac{1}{\xi T}\mathbb{E}\{F_L(Q_w(T))\\ &\leqslant \frac{\sigma}{\xi T}\sum_{t=1}^{T-1}\sum_{w=1}^{N} \mathbb{E}\{\gamma_w(t)a_w(t)\} + \frac{\varphi}{\xi}.\end{aligned} \tag{19}$$

Therefore, we prove that, by following the proposed crowdsourcing quality control mechanism, the *upper bound* on the time-averaged pending task queue lengths for workers is directly proportional to σ. Overall, a larger σ value will result in high accuracy of ratings for review comments, but longer delays in obtaining these ratings from crowdsourcing workers.

V. Experimental Evaluation



35/43 人认为此评论有用

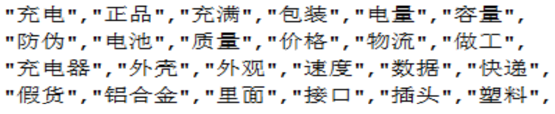
★☆☆☆☆ **while i am charging the power bank ,it is making noise.**

评论者[redacted] 于 2015年3月6日

尺寸: 16000mAh | 颜色: 银色 | 已确认购买

this is a product from a renowned company but the from the very first ,it is making disturbing noise whenever i am trying to recharge the power bank by any compatible adapter.i am so much disheartened..

Fig. 3. An example review from Amazon.cn.



"充电","正品","充满","包装","电量","容量",
"防伪","电池","质量","价格","物流","做工",
"充电器","外壳","外观","速度","数据","快递",
"假货","铝合金","里面","接口","插头","塑料",

Fig. 4. The keyword vector from our datasets.

In this section, we evaluate the performance of the proposed RREF using a dataset of real-world review comments crawled from the Amazon China e-commerce website.[3] The product reviews are written in Chinese and are obtained using a distributed web crawler over an 18 month period from March 2014 to August 2015. In total, we have obtained 10 datasets each containing 2,000 to 3,000 review comments under various sub-categories of mobile phone/communications products. An example review is shown in Figure 3.[4]. Using a word segmentation tool, we form a keyword vector from all comments (Figure 4).

RREF is compared against two state-of-the-art approaches: 1) the AdaBoost framework [22], and 2) JCF: the crowdsourcing enhanced classification framework proposed in [16]. We adopt the *average classification error rate*, $\overline{E} \in [0\%, 100\%]$, as the metric to compare the performance of RREF against these frameworks. The lower the value of $\overline{E}$, the better the performance.

A. Results and Discussions

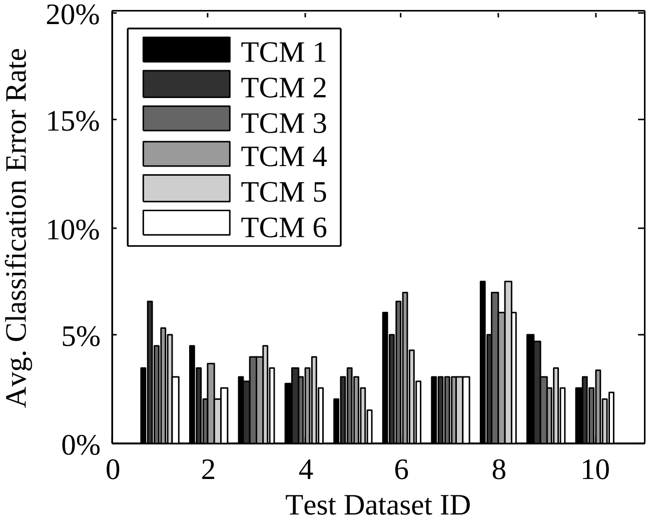


Fig. 5. $\overline{E}$ values of topic classification models (TCMs).

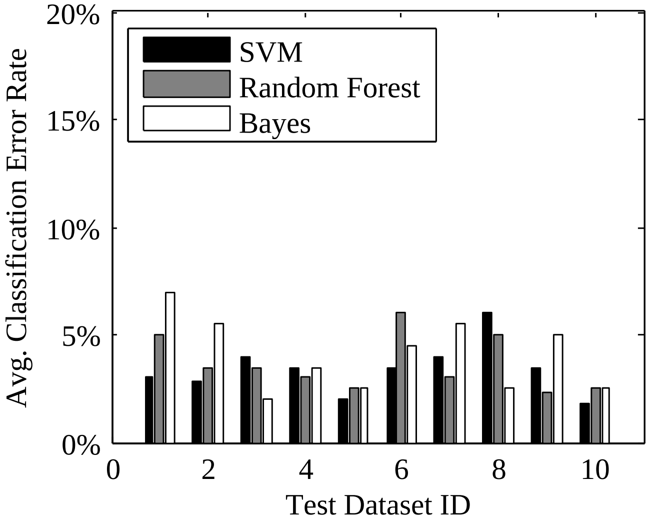


Fig. 6. $\overline{E}$ values of different classifiers on TCMs.

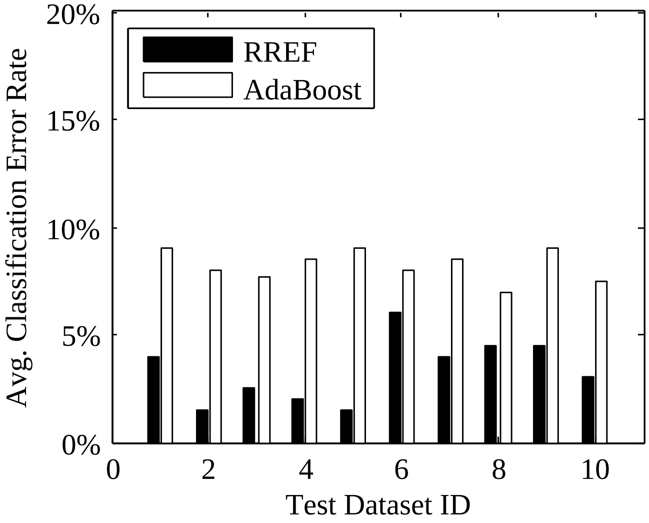


Fig. 7. RREF v.s. AdaBoost.

In the experiments, we include six different topic classification models (TCMs) and the SVM classifier, Random Forest classifier and the Naïve Bayes classifier for the proposed RREF to dynamically select from and adjust the weights. The performance of the six TCMs on the 10 test datasets we collected are shown in Figure 5. It can be observed that the TCMs perform differently as they each have different advantages and limitations. The same observations can be made for Figure 6 which shows the performance of the SVM, Random Forest and Naïve Bayes classifiers for the 10 test datasets.

Figure 7 compares the performance of RREF against that of AdaBoost. The principles of dynamic TCM selection and weight adjustment are the same in both frameworks. However, RREF is equipped with the additional features of review comment rating crowdsourcing with quality control (Algorithm 1) and keyword list reconstruction (Algorithm 2) based on TCM performance. The reconstructed keyword vector produced by RREF is shown in Figure 8. These two new features have resulted in improved performance by RREF. Over all 10 training sets (with over 20,000 review comments), RREF achieves an average classification error rate of 3.35%, which is 62% lower than the average classification error rate of 8.22% achieved by AdaBoost.

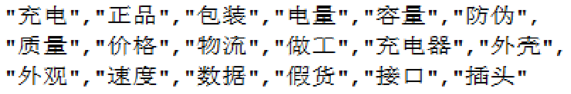

"充电","正品","包装","电量","容量","防伪",
"质量","价格","物流","做工","充电器","外壳",
"外观","速度","数据","假货","接口","插头"

Fig. 8. RREF reconstructed keyword vector.

[3] http://www.amazon.cn/

[4] The datasets used in our experiments can be downloaded from http://211.87.227.218:8081/en/download/

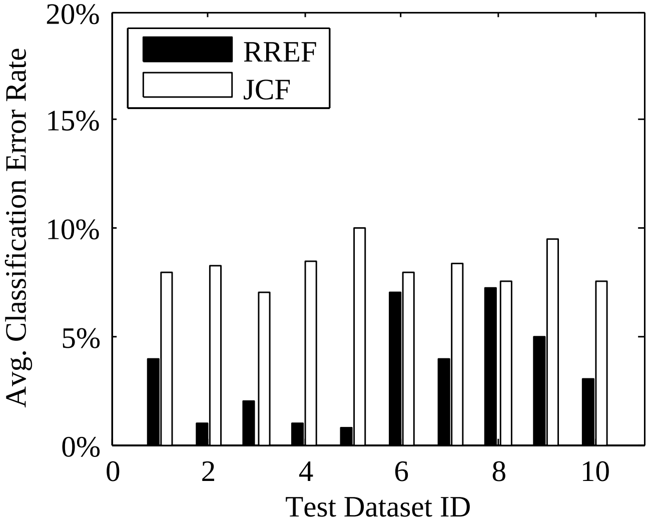


Fig. 9. RREF v.s. JCF.

Figure 9 shows the performance of RREF against that of JCF. JCF is closely related to the proposed RREF as it combines crowdsourced review comment ratings with active learning SVM classifiers with the aim to improve classification accuracy. Nevertheless, JCF does not possess any quality control mechanism for the crowdsourcing step, and does not adjust the weight values assigned to variants of the SVM classifiers. RREF achieves an average classification error rate of 3.5%, which is 58% lower than the average classification error rate of 8.27% achieved by JCF.

VI. Conclusions and Future Work

In this paper, we proposed a crowd intelligence powered dynamic machine learning framework - RREF - to help e-commerce systems accurately identify useful review comments from users. RREF contains a novel data-driven crowdsourcing quality control mechanism based on the *Lyapunov* queueing system optimization technique, and a dynamic classifier composition and weight adjustment approach with keyword list reconstruction. RREF offers an effective way to infuse crowd intelligence into machine learning to improve the overall review comment classification which is an important research areas in the field of NLP. Theoretical analysis proves the existence of performance bounds for the crowdsourcing quality control mechanism. Extensive experimental evaluations using a large scale real-world dataset demonstrated that RREF significantly outperforms state-of-the-art related frameworks.

In future research, we will further improve the crowdsourcing quality control mechanism to include other considerations, such as workers' social relationships [23], personal preferences [24] and miss-interpretation of outcomes [25], which may impact the biases in crowdsourced review comments ratings.

Acknowledgments

This research is supported, in part, by the National Natural Science Foundation of China under Grant No. 61572295, 61573212, 61272241; the Natural Science Foundation of Shandong Province of China under Grant No. ZR2014FM031, ZR2013FQ014; the Shandong Province Independent Innovation Major Special Project No. 2015ZDXX0201B03; the Shandong Province key research and development plan No. 2015GGX101015, 2015GGX101007; the Fundamental Research Funds of Shandong University No. 2015JC031; the National Research Foundation, Prime Minister's Office, Singapore under its IDM Futures Funding Initiative; and the Lee Kuan Yew Post-Doctoral Fellowship Grant.

References

[1] N. Zeichner, J. Berant, and I. Dagan, "Crowdsourcing inference-rule evaluation," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL'12)*, 2012, pp. 156–160.

[2] P. Michelucci and J. L. Dickinson, "The power of crowds," *Science*, vol. 351, no. 6268, pp. 32–33, 2016.

[3] L. von Ahn, B. Maurer, C. McMillen, D. Abraham, and M. Blum, "reCAPTCHA: Human-based character recognition via web security measures," *Science*, vol. 321, no. 5895, pp. 1465–1468, 2008.

[4] R. Silberzahn and E. L. Uhlmann, "Crowdsourced research: Many hands make tight work," *Nature*, vol. 526, pp. 189–191, 2015.

[5] Y. Yan, R. Rosales, G. Fung, and J. G. Dy, "Active learning from crowds," in *Proceedings of the 28 th International Conference on Machine Learning (ICML'11)*, 2015.

[6] J. Zhong, K. Tang, and Z.-H. Zhou, "Active learning from crowds with unsure option," in *Proceedings of the 24th International Conference on Artificial Intelligence (IJCAI'15)*, 2015, pp. 1061–1067.

[7] H. Yu, Z. Shen, C. Miao, and B. An, "Challenges and opportunities for trust management in crowdsourcing," in *Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'12)*, 2012, pp. 486–493.

[8] H. Yu, C. Miao, Z. Shen, C. Leung, Y. Chen, and Q. Yang, "Efficient task sub-delegation for crowdsourcing," in *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI-15)*, 2015, pp. 1305–1311.

[9] Y. Amsterdamer, Y. Grossman, T. Milo, and P. Senellart, "Crowd mining," in *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data (SIGMOD'13)*, 2013, pp. 241–252.

[10] H. Su, K. Zheng, J. Huang, H. Jeung, L. Chen, and X. Zhou, "Crowdplanner: A crowd-based route recommendation system," in *Proceedings of the IEEE 30th International Conference on Data Engineering (ICDE'14)*, 2014, pp. 1144–1155.

[11] J. Su, D. Xiong, Y. Liu, X. Han, H. Lin, J. Yao, and M. Zhang, "A context-aware topic model for statistical machine translation," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL'15)*, 2015, pp. 229–238.

[12] J. Derrac and S. Schockaert, "Inducing semantic relations from conceptual spaces: A data-driven approach to plausible reasoning," *Artificial Intelligence*, vol. 228, pp. 66–94, 2015.

[13] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research (JMLR)*, vol. 3, pp. 993–1022, 2003.

[14] D. Hovy, B. Plank, and A. Søgaard, "Experiments with crowdsourced re-annotation of a pos tagging data set," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL'14)*, 2014, pp. 377–382.

[15] S. Hao, C. Miao, S. C. Hoi, and P. Zhao, "Active crowdsourcing for annotation," in *Proceedings of the 2015 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT'15)*, 2015, pp. 1–8.

[16] J. Costa, C. Silva, M. Antunes, and B. Ribeiro, "On using crowdsourcing and active learning to improve classification performance," in *Proceedings of the 11th International Conference on Intelligent Systems Design and Applications (ISDA'11)*, 2011, pp. 469–474.

[17] P. Heymann and H. Garcia-Molina, "Turkalytics: analytics for human computation," in *Proceedings of the 20th International Conference on World Wide Web (WWW'11)*, 2011, pp. 477–486.

[18] Y. Wang and M. P. Singh, "Formal trust model for multiagent systems," in *Proceedings of the 20th International Joint Conference on Artifical Intelligence (IJCAI'07)*, 2007, pp. 1551–1556.

[19] H. Yu, Z. Shen, C. Leung, C. Miao, and V. R. Lesser, "A survey of multi-agent trust management systems," *IEEE Access*, vol. 1, no. 1, pp. 35–50, 2013.

[20] P. Schwartenbeck, T. H. B. FitzGerald, C. Mathys, R. Dolan, M. Kronbichler, and K. Friston, "Evidence for surprise minimization over value maximization in choice behavior," *Scientific Reports*, vol. 5, no. 16575, p. doi:10.1038/srep16575, 2015.

[21] M. J. Neely, *Stochastic Network Optimization with Application to Communication and Queueing Systems*. Morgan and Claypool Publishers, 2010.

[22] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[23] J.-P. Mei, H. Yu, Y. Liu, Z. Shen, and C. Miao, "A social trust model considering trustees' influence," in *Proceedings of the 17th International Conference on Principles and Practice of Multi-Agent Systems (PRIMA'14)*, 2014, pp. 357–364.

[24] H. Yu, Z. Shen, C. Miao, B. An, and C. Leung, "Filtering trust opinions through reinforcement learning," *Decision Support Systems (DSS)*, vol. 66, pp. 102–113, 2014.

[25] Y. Liu, S. Liu, H. Fang, J. Zhang, H. Yu, and C. Miao, "Reprev: Mitigating the negative effects of misreported ratings," in *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI-14)*, 2014, pp. 3124–3125.