Human Computation and Crowdsourcing: Works in Progress Abstracts:
An Adjunct to the Proceedings of the Third AAAI Conference on Human Computation and Crowdsourcing

Acquiring Reliable Ratings from the Crowd

Beatrice Valeri
University of Trento
valeri@disi.unitn.it

Shady Elbassuoni
American University of Beirut
se58@aub.edu.lb

Sihem Amer-Yahia
CNRS, LIG
sihem.amer-yahia@imag.fr

Abstract

We address the problem of acquiring reliable ratings of items such as restaurants or movies from the crowd. We propose a crowdsourcing platform that takes into consideration the workers' skills with respect to the items being rated and assigns workers the best items to rate. Our platform focuses on acquiring ratings from skilled workers and for items that only have a few ratings. We evaluate the effectiveness of our system using a real-world dataset about restaurants.

Introduction

In rating websites such as *Yelp* or *MovieLens*, some people provide untruthful ratings, either because they are cheating, or because they are not knowledgeable enough about the items they are rating. In addition, the number of ratings are usually not balanced across the items being rated. Some items have many ratings while others have a few ratings. Untruthful and imbalanced ratings can deteriorate recommendation accuracy in these rating websites.

In this paper, we present a novel crowdsourcing platform that acquires reliable ratings for a set of items from a set of workers. A reliable rating is a truthful rating provided by a skilled worker. Our data acquisition differs from a recommendation system in that it focuses on acquiring more data (meaning, ratings for items that only have a few ratings) and that it finds workers who are most likely to be knowledgeable about items (as opposed to workers who will discover items through the system, experience those items and then come back and rate them).

Most related work either assume the existence of one valid ground truth (Li, Zhao, and Fuxman 2014; Satzger et al. 2012; Karger, Oh, and Shah 2011; Ho, Jabbari, and Vaughan 2013), or are generally post-processing methods (Tian and Zhu 2012), or both (Joglekar, Garcia-Molina, and Parameswaran 2013; Wolley and Quafafou 2013). In our setting, rating of items is a subjective task and there is no single correct rating that can be used to estimate worker skills. Moreover, we do not use worker skills as a post-processing method as most previous work, but use it to dictate the choice of which items to rate by which workers.

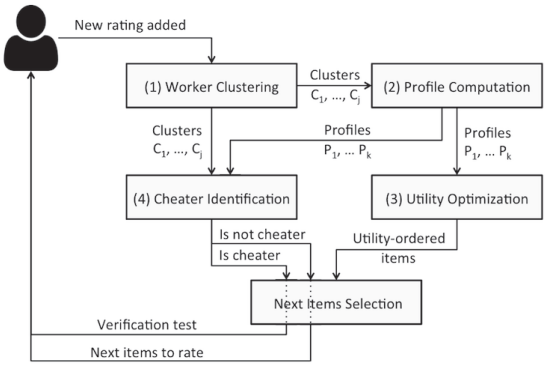

Figure 1: The Crowdsourcing Platform.

Crowdsourcing Platform

Our platform (shown in Figure 1) works as follows. First, it groups the items to be rated into a set of clusters based on their characteristics, which we refer to as itemsets. Itemsets are then used to model that workers are more skilled to rate certain types of items more than others. To represent worker skills, our framework associates each worker with a profile which is a vector of scores representing the skills of the worker for each itemset. The profile of a worker is constantly updated based on the ratio of items a worker has rated per itemset and the agreement of the worker with other *similar skilled* workers in the platform. To find groups of similar workers, we constantly cluster workers based on the ratings they provide using incremental hierarchical clustering. Workers who *consistently* fail to join any cluster, or have low profile values for *all* itemsets are suspected to be unskilled workers since they provide very different ratings from all other workers in the system. Such workers are then asked to pass a verification test which is simply another set of rating tasks where a worker is asked to rate items she has rated before. The verification test is designed as another rating task to disguise it from actual unskilled workers and to not turn off falsely-flagged workers. A worker passes the verification test if she is relatively consistent with her previous ratings, otherwise she is banned from the system.

Finally, our platform relies on a utility function to pro-

vide a given worker with the best items to rate. Our utility function is composed of two sub-components. The first component, $SetUtility(w, I_j)$ takes into consideration the worker profile and the number of ratings the worker has already provided for the itemset I_j. The second component, $ItemUtility(w, i)$, takes into consideration the number of ratings available for the item i and the closeness of the item to other items the worker knows.

More precisely, given a worker w and an itemset I_j, the $SetUtility$ component is defined as follows:

$$SetUtility(w, I_j) = \beta_1 \cdot (1 - \frac{\#ratings(w, I_j)}{MAX_k \#ratings(w, I_k)}) + \beta_2 \cdot (w.p_j)$$

where $\beta_1 + \beta_2 = 1$, $\#ratings(w, I_j)$ is the number of ratings the worker w provided for I_j and $w.p_j$ is the profile value of worker w for I_j. Similarly, given a worker w and an item i, the $ItemUtility$ component is defined as follows:

$$\begin{aligned} ItemUtility(w, i) &= \beta_3 \cdot (1 - \frac{\#ratings(i)}{MAX_j \#ratings(j)}) \\ &+ \beta_4 \cdot \frac{\Sigma_{j \in I_k^w} sim(i, j)}{|I_k^w|} \end{aligned}$$

where $\beta_3 + \beta_4 = 1$, $\#ratings(i)$ is the total number of ratings for item i and I_k^w is the set of items that worker w knows. The similarity $sim(i, j)$ is the similarity between two items i and j and it can be measured based on their characteristics (e.g, geographic distance between restaurants).

The final utility function $utility(w, i)$ of item i belonging to itemset I_j for worker w is then computed as the average of $ItemUtility(w, i)$ and $SetUtility(w, I_j)$. Once the utilities of every item for a given worker w are computed, we pick the item i for which $utility(w, i)$ is maximum and provide this item to the worker w to rate.

Evaluation

To test the effectiveness of our platform in acquiring reliable ratings, we use a real dataset of restaurant ratings, identify skilled and unskilled workers in this dataset using our platform, and measure the errors made by an off-the-shelf recommendation system (Lee, Sun, and Lebanon 2012).

Using our platform, we acquired a total of 540 ratings for 50 selected restaurants in Grenoble, France. The ratings were collected from 57 workers, seven of which were skilled workers and 10 were unskilled workers with random ratings.

We split our dataset into training and test sets. For each rating in the test set, we computed its predicted value based on the training set and then calculated the RMSE (root mean squared error). Using the full dataset, containing both skilled and unskilled workers, we got an RMSE of *2.202*. Using only skilled workers, we obtained an RMSE value of *1.021*. We can therefore conclude that the ability to isolate unskilled workers in this dataset reduced recommendation error by *53.6%*. This result is quite promising and shows the effect of worker skill on the quality of the ratings acquired.

We also compared our utility function to two other baseline utility functions: i) a recommendation-based utility function, in which the next item shown to the worker is the one recommended to the worker according to the ratings she already gave using an off-the-shelf recommendation system, and ii) a random utility function, in which the next item is randomly selected. To test which utility function performs best, we analyzed the number of ratings the framework asked each unskilled worker before identifying her. On average, the random utility function needed to show *36* items before identifying an unskilled worker and the recommendation-based utility function needed to show *34* items before identifying an unskilled worker, while our utility function needed to present only *25* items before identifying an unskilled worker. This means that our utility function was able to identify unskilled workers earlier than the other two functions by at least *30%*.

Conclusion

We presented a crowdsourcing platform to acquire reliable ratings of items. Our platform relies on incremental hierarchical clustering to estimate worker skills and a carefully-designed utility function to assign underexposed items to the most skilled workers. We have demonstrated the effectiveness of our platform using a real dataset about restaurants. In the future, we plan to run more experiments on other datasets and to extend our framework to identify worker bias when rating items.

References

Ho, C.; Jabbari, S.; and Vaughan, J. W. 2013. Adaptive task assignment for crowdsourced classification. In *Proceedings of the 30th International Conference on Machine Learning*.

Joglekar, M.; Garcia-Molina, H.; and Parameswaran, A. 2013. Evaluating the crowd with confidence. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Karger, D. R.; Oh, S.; and Shah, D. 2011. Budget-optimal task allocation for reliable crowdsourcing systems. *CoRR* abs/1110.3564.

Lee, J.; Sun, M.; and Lebanon, G. 2012. A comparative study of collaborative filtering algorithms. *arXiv preprint arXiv:1205.3193*.

Li, H.; Zhao, B.; and Fuxman, A. 2014. The wisdom of minority: Discovering and targeting the right group of workers for crowdsourcing. In *Proceedings of the 23rd International Conference on World Wide Web*.

Satzger, B.; Psaier, H.; Schall, D.; and Dustdar, S. 2012. Auction-based Crowdsourcing Supporting Skill Management. *Information Systems, Elsevier*.

Tian, Y., and Zhu, J. 2012. Learning from crowds in the presence of schools of thought. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Wolley, C., and Quafafou, M. 2013. Scalable expert selection when learning from noisy labelers. In *Proceedings of the 2013 12th International Conference on Machine Learning and Applications*.