Improving Performance by Re-Rating in the Dynamic Estimation of Rater Reliability

Alexey Tarasov ALEKSEJS.TARASOVS@STUDENT.DIT.IE
Sarah Jane Delany SARAHJANE.DELANY@DIT.IE
Brian Mac Namee BRIAN.MACNAMEE@DIT.IE
Applied Intelligence Research Centre, Dublin Institute of Technology, Kevin St., Dublin 8, Ireland

Abstract

Nowadays crowdsourcing is widely used in supervised machine learning to facilitate the collection of ratings for unlabelled training sets. In order to get good quality results it is worth rejecting results from noisy/unreliable raters, as soon as they are discovered. Many techniques for filtering unreliable raters rely on the presentation of training instances to the raters identified as most accurate to date. Early in the process, the true rater reliabilities are not known and unreliable raters may be used as a result. This paper explores improving the quality of ratings for training instances by performing re-rating. The re-rating relies on the detection of such instances and the acquisition of additional ratings for them when the rating process is over. We compare different approaches to re-rating and compare the improvements in labeling accuracy and the labeling costs of these approaches.

1. Introduction

Crowdsourcing, a type of participative activity where a task is proposed to a group of individuals (Estellés-Arolas & Gonzalez-Ladron-de Guevara, 2012), is widely used to collect ratings for data to be used in supervised machine learning (Ambati et al., 2010; Brew et al., 2010; Snow et al., 2008). The usual scenario is that each training instance is presented to several raters whose results are then combined to produce a single *target rating*.

Raters usually have different expertise and degree of commitment, and as a result submit ratings of varying accuracy. Low accuracy ratings can have a significant negative impact on the quality of target ratings. Therefore, unreliable raters should be detected and their ratings disregarded. There are two ways to use rater reliability in the calculation of target ratings: *static* and *dynamic*. Static approaches first gather all ratings in an undetermined order and then calculate rater reliability and target ratings (Raykar et al., 2010; Whitehill et al., 2009). The target rating is typically the average of the ratings weighted by the calculated reliability of the rater. Dynamic approaches, which are the focus of this paper, present training instances one by one, tracking rater reliability while raters provide ratings (Donmez et al., 2009; Welinder & Perona, 2010; Yan et al., 2011). In dynamic approaches, reliabilities are estimated as ratings are gathered, and each training instance is presented only to those raters who are deemed to be reliable, based on previously rated training instances. Thus, unreliable raters are discovered dynamically, in contrast to static approaches which wait until the end of the rating process to calculate rater reliability.

The dynamic selection of the most accurate raters is an example of a problem where a trade-off between exploration and exploitation has to be found. Exploration involves giving enough chances to all raters, even if some of them later prove unreliable, in order to precisely estimate their reliabilities. When the reliabilities are known, exploitation can begin. During exploitation only raters considered to be reliable are asked to rate. Thus, dynamic estimation of rater reliability inevitably leads to the collection of ratings from inaccurate raters. The probability of asking inaccurate raters is especially high in the beginning of the rating process, during the exploration phase. It means that training instances rated at the beginning have a higher probability of having noisy target ratings.

This paper explores how to reduce the error of target

ICML Workshop: Machine Learning Meets Crowdsourcing, Atlanta, Georgia, USA, 2013.

ratings for training instances rated at the exploration stage. At the end of the rating process the top raters are selected and requested to supply additional ratings for a certain portion of training instances rated at the beginning of the process. When new ratings are acquired, the old ratings are dropped. Now that raters known to be reliable had been asked, the error in the training instances target ratings decreased. We tested two approaches to selecting the number of training instances to be re-rated: (i) a fixed approach, which re-rated the initial x% of training instances, and (ii) a trend-based approach, which used trend analysis techniques to find the boundary between exploration and exploitation in the rating process. The dynamic rating approach used in this work involves *multi-armed bandits* (MABs) (Tarasov et al., 2012).

The paper is structured as follows: Section 2 describes related work. The experiment's methodology is covered in Section 3. Section 4 presents the results and discusses them, while Section 5 concludes the paper.

2. Related work

Crowdsourced rating of corpora for supervised machine learning is widely used in a variety of domains, including machine translation (Ambati et al., 2010) and sentiment analysis (Brew et al., 2010). Generally, a fixed payment is made for each rating collected. One of the main challenges of crowdsourced rating of corpora is to get the target ratings as accurate as possible while paying as little as possible for them. In any application area some raters might complete the task without fully engaging in it (Downs et al., 2010) which results in inaccurate ratings. Such unreliable raters can have a significant negative impact on target ratings (Whitehill et al., 2009). There are a number of different techniques that can be used to reduce this negative effect which can be divided into three categories based on the stage of the rating process at which they are applied:

1. Before the rating process starts: only raters who successfully complete a qualification task rate training instances (Downs et al., 2010). Usually, such a task involves rating a few training instances for which the correct ratings are already known.

2. After the rating process finishes (static methods): all training instances are presented for rating simultaneously, and each rater can rate as many of them as desired. The process usually finishes when a certain number of ratings for each training instance is gathered. Then an expectation maximisation algorithm is used to calculate both rater reliabilities and target ratings (Dekel & Shamir, 2009; Raykar et al., 2010; Whitehill et al., 2009). Target ratings are usually calculated such that ratings from less reliable raters have smaller weights than those from reliable ones.

3. During the rating process itself (dynamic methods): training instances are presented one by one (or in small batches) to a subset of raters. The reliability of raters is tracked dynamically as they rate training instances, the calculation of target ratings also happens as ratings progress (Donmez et al., 2009; Welinder & Perona, 2010; Yan et al., 2011).

Currently there is no strong evidence in the literature that methods from the first group are actually beneficial. Heer & Bostock (2010) report that these methods were able to reduce the proportion of invalid ratings from 10% to 0.4%, while Su et al. (2007) were unable to find any correlation between rater performance in the qualification task and in rating of actual training instances. At the same time, both static and dynamic techniques are widely used and have been proven to be successful (Donmez et al., 2009; Raykar et al., 2010). According to Welinder & Perona (2010), dynamic techniques should be preferred over static ones if it is important to bring down the total cost of the rating process. A dynamic technique makes use of all ratings gathered, while a static approach does not. This is because ratings that have already been paid for may have come from inaccurate raters and be discarded while calculating the target ratings.

In general, dynamic approaches to estimating rater reliability use either probabilistic frameworks (Welinder & Perona, 2010; Yan et al., 2011) or MABs (Donmez et al., 2009; Tarasov et al., 2012). Probabilistic frameworks make certain assumptions, e.g., prior beliefs about the rater reliability and the statistical distribution of rater errors. Such assumptions might not always hold true in real-life tasks, which can have a negative impact on the results of using these approaches. In contrast, MABs do not make such assumptions and therefore can be used for any task at hand.

An MAB represents a problem of selection between k alternatives as a k-armed gambling machine (Vermorel & Mohri, 2005). In our previous work (Tarasov et al., 2012) we explored whether they can be used for dynamic estimation of rater reliability. We found that two algorithms—KL-UCB and ϵ-first—usually lead to target ratings of higher accuracy compared to those produced by IEThresh (which also is an MAB, although it is not acknowledged directly (Donmez et al., 2009)). In our experiments we determined the number of raters to be asked to rate each training instance

(N) in advance. At each iteration, or *round*, a single training instance was rated. Each rater who rated this instance received a *reward*, a number which represented how close the rating of this rater was to the consensus rating.

The ϵ-first algorithm (Vermorel & Mohri, 2005) divides all rounds into two phases: exploration and exploitation. At the exploration phase N random raters are asked to rate each training instance, which ensures they were all given an equal chance to show their performance. When exploitation starts, raters who have the best performance to date are asked to rate. The reliability score of a rater is simply an average of his rewards to date. The ϵ-first algorithm has one parameter, ϵ, which determines the proportion of training instances to be rated at the exploration stage.

In contrast, KL-UCB (Garivier & Cappe, 2011) does not divide all rounds into exploratory or exploitative like ϵ-first. In KL-UCB the reliability score of a rater is a function of rewards received by him to date, and this function depends on the number of training instances the rater has rated. If a rater has a high reliability score, it means that either (i) he has rated many training instances and showed himself to be a reliable rater, or (ii) he has rated few training instances so we are not sure about his reliability yet. At any time, the rater with a high reliability score should be asked to rate.

In all the dynamic approaches described above, collection of ratings from unreliable raters inevitably happens at the beginning of the process. This can lead to a situation where a certain part of the training set has noisy target ratings. This issue is not covered by state-of-the-art research in crowdsourcing to the best of our knowledge, and we are not aware of any approaches to correcting this exploration-stage error in other applications areas, where a trade-off between exploration and exploitation is required.

In the next section we propose a few strategies for compensating for these errors and suggest how the performance of these approaches can be measured.

3. Methodology

The goal of our experiment is to compare different strategies for increasing the accuracy of target ratings in dynamic approaches. We simulated the crowdsourced collection of ratings; instead of asking real people to rate training instances, we used datasets where a number of raters had already rated all training instances. When we required a rating from a certain rater, instead of querying a real rater through the Internet, we simply took the rating from the dataset.

This section describes the experiment we carried out, the re-rating approaches we propose, the datasets we used and explains performance measures.

3.1. Experiment

The collection of ratings in our experiment consisted of two stages:

1. Initial rating process, during which MABs were used to dynamically estimate rater reliability. The initial rating process finished when every training instance had been presented to raters and rated.

2. Re-rating: we detected which training instances were rated at the exploration phase and selected the most reliable raters at the end of the initial rating process. To improve the accuracy of the target ratings of these training instances, we collected replacement ratings for these instances from the most reliable raters.

Three corpora were used for our experiments:

The Vera am Mittag (VAM) corpus (Grimm et al., 2008), which contains non-acted video recordings of a talk show, divided into short segments. Each speech segment is rated on three continuous dimensions. Ratings on all dimensions are in the [-1, -0.5, 0, 0.5, 1] set. The three dimensions are activation (how active or passive the recording is), evaluation (how positive or negative it is) and power (how dominant the speaker is). We used a portion of the VAM corpus containing 478 speech recordings, each rated by same 17 raters. We formed three separate datasets, one for each emotional dimension.

Our initial experiments revealed that the performance of all raters in all three emotional datasets was very similar. Thus, no matter which raters were chosen, the resulting target ratings were of the same high quality. To introduce some variability into the ratings we added 10 additional noisy raters to each dataset. The ratings for these noisy raters were generated by adding a random noise term, from a Guassian distribution, to the actual rating of each recording, similar to the approach adopted by Raykar et al. (2010).

The Jester dataset[1] containing 4.1 million continuous ratings (on the scale [-10, 10]) of 100 jokes rated by 73,421 people. Each joke is rated by a varying number of raters. A subset of ratings from 20 raters who have rated all 100 jokes was used as the experimental dataset.

[1] http://goldberg.berkeley.edu/jester-data/

The MovieLens 10M dataset[2] consisting of 10 million ratings across 10,000 movies by 72,000 users. As in Jester, each movie is rated by a different number of people by assigning a rating in a [1, 5] range. We extracted a subset of 288 movies, each rated by the same 20 raters for our experiments.

As no true target ratings were available for any of the datasets, we calculated them using the approach of Raykar et al. (2010). We refer to these as the *gold standard* ratings.

We conducted the initial rating process, the goal of which was to rate T training instances using two different MAB approaches: ϵ-first and KL-UCB. The initial rating process consisted of the following steps:

1. Select a training instance to be rated: training instances were presented to raters one by one. In order to determine the order of presentation, active learning was used for three VAM datasets. Active learning is a semi-supervised machine learning approach that can be used to build accurate classifiers and predictors from collections of unrated data, with minimal rating effort. This is achieved by only rating those instances from a large pool that are deemed, using a selection strategy, to be most informative. We used the active learning approach of Burbidge et al. (2007) and a deterministic clustering approach to seed this process (Hu et al., 2010). If a few training instances had the same "informativeness" score, one of them was chosen at random. Active learning was not possible for the Jester and MovieLens datasets as no features are available for the instances so we presented training instances from those datasets in random order.

2. Select raters: The most reliable N raters were asked to rate the training instance selected during the previous step. Ties in rater reliability scores were broken randomly, as were ties in step 1.

3. Calculate the target rating: we used an average of the N ratings received for a training instance as the estimated target rating for the training instance.

4. Update the rater reliabilities: the closer the rating given by a rater was to the estimated target rating, the more reliable the rater. The reward for a rater was a normalised inverse absolute difference between the estimated target rating (calculated at step 3) and the rating provided by the rater (received at step 2) and was in the [0, 1] interval. All rewards received by a rater were stored and used to calculate his reliability score. In ϵ-first, this score was just an average of all rewards received by the rater to date, while in KL-UCB it was a function of the number of rewards received to date and their values (Garivier & Cappe, 2011).

5. Repeat from step 1 until all training instances are rated.

When the initial rating process was finished, the acquisition of additional ratings proceeded as follows:

1. Training instances for which the acquisition of additional ratings is required were selected.

2. Additional ratings for the instances were solicited from the N most reliable raters (as at the end of the initial rating process) to replace those previously collected. In the situation when a rater had already provided a rating for a given training instance, we simply reused the old rating. Collecting a new rating would amount to "purchasing" multiple ratings for the same training instance from the same individual.

3. The target ratings for the selected assets were updated. A new target rating was an average of newly acquired ratings.

We conducted different runs of the experiment for each dataset, varying the number of raters asked to rate every training instance $N = 3, 5, 7, 9, 11, 13, 15$.

Different approaches to re-rating covered in this paper differ in terms of how step 1 was executed. We used two baselines and two re-rating approaches in our experiments. The baselines were as follows:

None: no re-rating happened at all (zero training instances were selected for re-rating).

Full: all training instances rated during the initial rating process were re-rated.

We considered the following re-rating approaches:

Fixed: the first x% of the training instances rated in the initial rating process were selected for re-rating.

Trend-based: the number of training instances to be re-rated was not set in advance and was determined via trend analysis. We assumed that there exists a border between exploration and exploitation in the initial rating process. When this border was found, only training instances rated at the exploration phase were re-rated.

As discussed, noisy ratings are often collected during the exploration phase. This means that target ratings for the training instances rated at this phase may be unreliable. As the exploration stage progresses, rater reliabilities are learned and unreliable raters are cho-

[2] http://www.grouplens.org/node/73

sen more and more rarely. The error in target ratings for the exploration stage exhibits the negative trend. Finally, when rater reliabilities are estimated well enough, only reliable raters are asked, and the error of target ratings remains stable and relatively low.

In practice, true target ratings are not known during the rating process. Consequently, they can not be used in locating the start of the exploitation. However, the standard deviation (SD) of N ratings received for each training instance can be used as a proxy measure, as reliable raters tend to agree with one another. Our initial experiments revealed that the behaviour of the SD was generally similar to that of the error making it a suitable proxy for the error.

First, we started with the sequence of all SDs $(\sigma_1, \sigma_2, ..., \sigma_T)$ for ratings received in the initial rating process, where T was the total number of training instances. Each σ_i was the SD of ratings which selected raters supplied for the training instance rated at i-th round. The Mann-Kendall test was used to check if there was a negative trend in this sequence. If there was a negative trend, the first SD was removed and the check for trend was performed again on the $(\sigma_2, \sigma_3, ..., \sigma_T)$ sequence. The process continued until SDs in the sequence did not exhibit a negative trend. This process is illustrated in Figure 1.

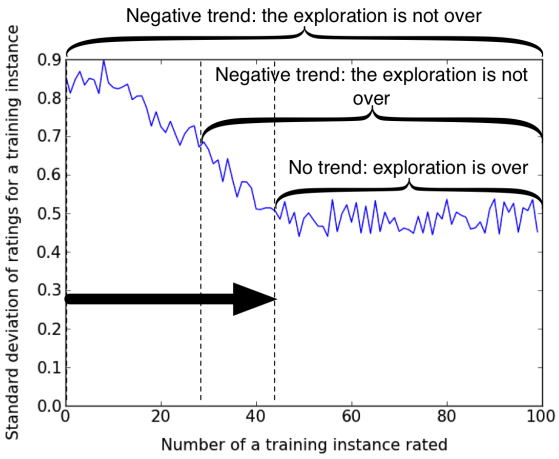

Figure 1. Detection of the boundary between exploration and exploitation using trend analysis. In this example, all training instances rated before the training instance #43 were rated at the exploration stage and had to be re-rated.

3.2. Performance measures

Two performance measures are important for each re-rating approach: (i) the cost of ratings C, expressed as a number of acquired ratings and (ii) the error of the resulting target ratings E, which is the average absolute difference between target ratings predicted by raters and the gold standard ratings, measured as a percentage of the full rating scale. For instance, if the error is 0.25 on MovieLens dataset, where the width of the full [1,5] scale is four, $E = 0.25/4 = 0.0625 = 6.25\%$. The value of C includes both ratings gathered during the initial rating process and during re-rating. A good approach should lead to low values of both C and E.

In order to compensate for possible ties at steps 1 and 2 of the initial rating process, we conducted the experiment for each emotional dataset and each N five times, and reported averaged values of C and E. In MovieLens and Jester we did 50 runs instead of five, because a large number of ties between training instances introduced a significant random component to the order of presentation of training instances.

In our experiments, we had to rank different re-rating approaches. This task is often encountered in machine learning and is usually relatively simple. For instance, if different classification techniques are compared, they usually are ranked according to their accuracy (Demsar, 2006). However, when more than one performance measure is used, more sophisticated techniques have to be applied. Estimating how well each re-rating approach meets both low C and low E criteria is a multi-criteria decision analysis (MCDA) problem. Many MCDA techniques exist and are widely used in different application areas, however, according to Toloie-Eshlaghy & Homayonfar (2011), the most popular of these are members of the analytic hierarchy process (AHP) family. Triantaphyllou & Baig (2005) performed the analysis of different AHP methods and strongly recommended using a *multiplicative AHP* (MAHP). We rank re-rating approaches using the approach by Triantaphyllou & Baig (2005), applying MAHP with the two criteria being C and E. The MAHP also allows some criteria to be more important than others by assigning them different weights, which should sum up to 1. In our experiments, both error and cost were equally important, so both had equal weight $W_C = W_E = 0.5$. The inputs to the MAHP are four pairs of C and E, one for each re-rating approach. The result of the MAHP are ranks from 1 to 4 assigned to four re-rating approaches. Thus, the final result of the experiment was 70 rankings (2 MABs $\times$ 5 datasets $\times$ 7 values of N) of four re-rating approaches.

We used the Friedman test following Demsar (2006) to determine if there was a statistically significant difference between at least two approaches and the Bergmann-Hommel post-hoc test to discover which approaches do differ (Garcia & Herrera, 2008). The re-

sult of this statistical testing was a grouping of approaches such that there was no statistically significant difference between approaches from the same group.

4. Results and discussion

With ϵ-first, in 30 out of 35 experiments the trend-based approach re-rated between 10% and 10.1% of training instances. Therefore, the border between exploration and exploitation usually lay in the region of $\epsilon = 0.1$. Figure 2 shows just one SD graph as an illustration, but very similar behaviour was observed in other experiments as well. Such a crisp distinction between exploration and exploitation was not very surprising, as ϵ-first explicitly explores during first $\epsilon \cdot T$ rounds and exploits the rest of the time. A comparison of average ranks concluded that there was a statistically significant difference between re-rating approaches (Friedman test p-value <0.001), and the following two groups were identified by the Bergmann-Hommel test with significance at $\alpha = 0.05$ level[3] (average ranks are given in parentheses):

1. Fixed, $x = 10$ (2.00) and trend-based (2.14)
2. None (2.85) and full (3.00)

None and full were the worst approaches, while fixed and trend-based turned out to be the best. As the boundary between exploration and exploitation was almost always at about 10%, the fixed approach worked well, always re-rating 10% of training instances. Although there is no statistically significant difference between fixed and trend-based approaches, we would recommend the use of fixed as a simpler alternative, when ϵ-first is used to select raters dynamically.

In our experiments, the first $\epsilon \cdot T$ training instances had an average error of 6.57%, while the rest had an error of 3.34%. When initial training instances were re-rated using fixed re-rating, the error on those instances dropped to 3.44%, i.e. halved. The initial rating process required 3,225.6 ratings on average; re-rating required an additional 179.26 ratings on average.

With regard to the significance of the reduction of 3.13% in the average error, we would like to cite Wagstaff (2012), who notes that a decrease in error of the same absolute value can have a completely different meaning and impact, depending on the area. Indeed, some application areas such as medical diagnosis or biometrics would be intolerant even to small mistakes. Even if an incorrect decision is taken in 1% of cases, it still results in a lot of mis-diagnoses or security breaches. There is little need for re-rating if the error of a few percent is tolerable.

[3]The results reported below use $\alpha = 0.05$, unless otherwise specified.

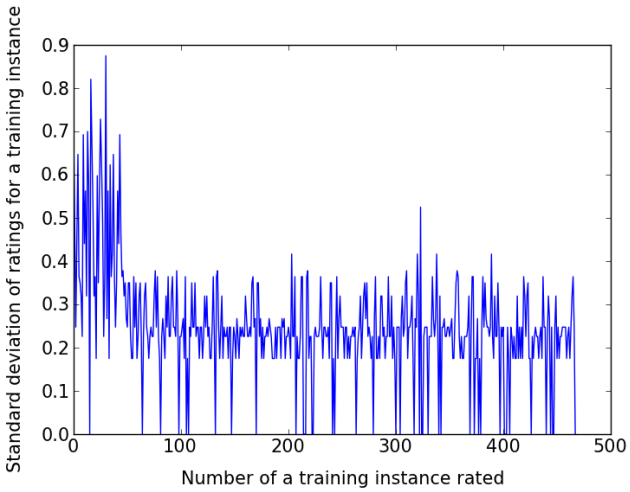

Figure 2. Change in SD of ratings, as training instances are being rated (*Activation*, $N = 3$).

With KL-UCB, the border between exploration and exploitation on VAM datasets lay in quite a wide range from 5% to 50%, unlike ϵ-first, where this border was almost always around 10%. The typical SD graph for MovieLens and Jester did not exhibit any trends or phases. One possible explanation is that Jester and MovieLens represent very subjective problems, where it is difficult to find a subset of raters tending to agree with one another. In order to investigate this, we calculated average absolute errors for all raters in all five datasets (Figure 3). Red crosses represent noisy, artificially generated raters who do not agree, as their ratings were generated independently of one another. The errors of raters who were in the original datasets (blue crosses) is spread uniformly in all datasets, but the range of rater reliabilities is bigger on MovieLens and Jester than on emotional datasets. This means that, in general, raters in MovieLens and Jester tended to disagree with each other more than on emotional datasets. Thus, it was more difficult for KL-UCB to pick a set of raters who agree, which explains the absence of trend in SD graphs. Nevertheless, the results of our previous work (Tarasov et al., 2012) strongly suggest that even on these more challenging datasets, KL-UCB still can pick reliable raters quite successfully.

We launched the fixed approach with $x = 50$ in order to ensure that it always re-rates all exploration-phase training instances. The approaches were split into three groups (Friedman p-value <0.001):

1. None (1.74), trend-based (2.09)
2. Fixed, $x = 50$ (2.66)
3. Full (3.51)

The trend-based approach often re-rated only a few initial training instances and therefore did not produce a big difference in cost or error, compared to no re-rating. When the fixed approach was used, the average error over the initial 50% of training instances changed from 5.54% (no re-rating) to 4.67%. However, this decrease in error required significant additional costs: compared to an average $C = 3240.8$ for no re-rating, the fixed approach resulted in $C = 4021.23$ (an increase of 24.1%). This poor "value for money" resulted in the fixed approach ranking worse than no re-rating.

The grouping did not change when we used $x = 25$ instead of 50 in the fixed approach (Friedman p-value <0.001):

1. None (1.77), trend-based (2.06)
2. Fixed, $x = 25$ (2.66)
3. Full (3.51)

Again, the significant additional cost resulted in a 1.04% decrease in error for the initial 25% of training instances, from 5.75% to 4.71%. We also tried several different values of x in the fixed approach, but the grouping remained the same.

We are interested in decreasing the error by re-rating training instances, however the increase in cost associated with re-rating should not be high. Our assumption was that both cost and error are equally important, and the weights of both cost and error were set to 0.5 while calculating the MAHP performance measure. However, there are some tasks (e.g. from medical or security applications) where we might be interested in bringing the error down, even if it costs a lot. We modelled such a task by setting $W_C = 0.05$ and $W_E = 0.95$ and compared the average ranks of full, none, trend-based and fixed (both $x = 25$ and $x = 50$) approaches. The resulting rankings and groupings are different to those received for equal weights (Friedman p-value <0.001, $\alpha = 0.1$ for Bergmann-Hommel was used):

1. Fixed, $x = 50$ (2.00) and full (2.11)
2. Fixed, $x = 25$ (2.91)
3. Trend-based (3.83) and none (4.14)

This ranking suggests that when accuracy is more important than cost, re-rating should be recommended. As before, the trend-based approach was not significantly different from no rating. However, the ranks of fixed and full approaches changed. It should be noted that re-rating 50% of the training instances proved to be as beneficial as doing a full re-rating. Re-rating more than 50% did not prove to be worthwhile. As discussed above, the exploration phase always finished by the time 50% of all training instances had been rated. This means that by that time KL-UCB had learned enough about rater reliability, and asked reliable raters to rate the second half of training instances. Thus, for the situation where improving the accuracy of the ratings is more important than the cost of getting ratings, based on these results, the re-rating of 50% of training instances can be recommended.

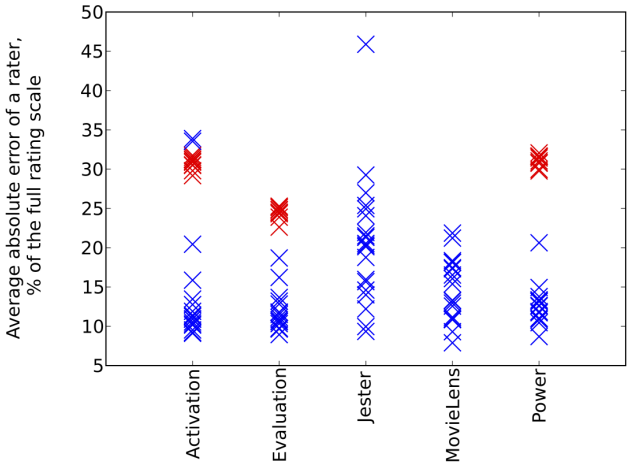

Figure 3. Mean errors of raters. Red crosses represent artificially generated, noisy raters, while blue crosses correspond to raters originally present in datasets. It is difficult to find a subset of raters who agree with each other in MovieLens and Jester.

5. Conclusions

The experiments reported in this paper show that re-rating can indeed increase the accuracy of target ratings in crowdsourced rating of training sets when MABs are used to dynamically estimate rater reliability. For ϵ-first, both trend-based and fixed re-rating were the best, but we would recommend using the latter as a simpler alternative. In KL-UCB re-rating is advised only for tasks where error has much higher priority than cost, i.e. it is worthwhile to pay a lot even for a relatively small increase in accuracy. For such tasks, the re-rating of the first 50% of training instances worked well in our experiments.

Acknowledgments

This work was supported by Science Foundation Ireland under Grant No. 09-RFP-CMS253. We would like to thank Dr Rong Hu for providing the code for the deterministic clustering approach used to seed the active learning process, and Mr Stephen Kidney for help with proofreading the paper.

References

Ambati, V., Vogel, S., and Carbonell, J. Active Learning and Crowd-sourcing for Machine Translation. In *Procs of LREC*, 2010.

Brew, A., Greene, D., and Cunningham, P. Using Crowdsourcing and Active Learning to Track Sentiment in Online Media. In *Procs of ECAI*, 2010.

Burbidge, R., Rowland, J.J., and King, R.D. Active Learning for Regression Based on Query by Committee. *LNCS, IDEAL 2007*, 4881:209–218, 2007.

Dekel, O. and Shamir, O. Good Learners for Evil Teachers. In *Procs of ICML*, 2009.

Demsar, J. Statistical Comparisons of Classifiers over Multiple Data Sets. *Journal of Machine Learning Research*, 7:1–30, 2006.

Donmez, P., Carbonell, J.G., and Schneider, J. Efficiently Learning the Accuracy of Labeling Sources for Selective Sampling. In *Procs of KDD*, 2009.

Downs, J.S., Holbrook, M.B., Sheng, S., and Cranor, L.F. Are Your Participants Gaming the System? Screening Mechanical Turk Workers. In *Procs of CHI*, 2010.

Estellés-Arolas, E. and Gonzalez-Ladron-de Guevara, F. Towards an Integrated Crowdsourcing Definition. *Journal of Information Science*, March 9, 2012.

Garcia, S. and Herrera, F. An Extension on "Statistical Comparisons of Classifiers over Multiple Data Sets" for all Pairwise Comparisons. *Journal of Machine Learning Research*, 9:2677–2694, 2008.

Garivier, E. and Cappe, O. The KL-UCB Algorithm for Bounded Stochastic Bandits and Beyond. In *Procs of COLT*, 2011.

Grimm, M., Kroschel, K., and Narayanan, S. The Vera am Mittag German Audio-Visual Emotional Speech Database. In *Procs of ICME*, 2008.

Heer, J. and Bostock, M. Crowdsourcing Graphical Perception: Using Mechanical Turk to Assess Visualization Design. In *Procs of CHI*, 2010.

Hu, R., Mac Namee, B., and Delany, S.J. Off to a Good Start: Using Clustering to Select the Initial Training Set in Active Learning. In *Procs of FLAIRS*, 2010.

Raykar, V.C., Yu, S., Zhao, L.H., Valadez, G.H., Florin, C., Bogoni, L., and Moy, L. Learning From Crowds. *Journal of Machine Learning Research*, 11: 1297–1322, 2010.

Snow, R., O'Connor, B., Jurafsky, D., and Ng, A. Y. Cheap and Fast—But Is It Good? Evaluating Non-expert Annotations for Natural Language Tasks. In *Procs of EMNLP*, 2008.

Su, Q., Pavlov, D., Chow, J.H., and Baker, W.C. Internet-scale Collection of Human-reviewed Data. In *Procs of WWW*, 2007.

Tarasov, A., Delany, S.J., and Mac Namee, B. Dynamic Estimation of Rater Reliability in Regression Tasks using Multi-Armed Bandit Techniques. In *Workshop on Machine Learning in Human Computation and Crowdsourcing at ICML*, 2012.

Toloie-Eshlaghy, A. and Homayonfar, M. MCDM Methodologies and Applications: A Literature Review from 1999 to 2009. *Research Journal of International Studies*, 21:86–137, 2011.

Triantaphyllou, E. and Baig, K. The Impact of Aggregating Benefit and Cost Criteria in Four MCDA Methods. *IEEE Transactions on Engineering Management*, 52(2):213–226, 2005.

Vermorel, J. and Mohri, M. Multi-armed Bandit Algorithms and Empirical Evaluation. *LNAI, Machine Learning: ECML 2005*, 3720:437–448, 2005.

Wagstaff, Kiri. Machine Learning that Matters. In *Procs of ICML*, 2012.

Welinder, P. and Perona, P. Online Crowdsourcing: Rating Annotators and Obtaining Cost-effective Labels. In *Workshop on Advancing Computer Vision with Humans in the Loop at CVPR*, 2010.

Whitehill, J., Ruvolo, P., Wu, T., Bergsma, J., and Movellan, J. Whose Vote Should Count More: Optimal Integration of Labels from Labelers of Unknown Expertise. In *Procs of NIPS*, 2009.

Yan, Y., Rosales, R., Fung, G., and Dy, J. Active Learning from Crowds. In *Procs of ICML*, 2011.