Quality Assurance in Crowdsourcing via Matrix Factorization based Task Routing

Hyun Joon Jung
«Supervised by Matthew Lease»
School of Information
University of Texas at Austin
{hyunJoon, ml}@utexas.edu

ABSTRACT

We investigate a method of crowdsourced task routing based on matrix factorization. From a preliminary analysis of a real crowdsourced data, we begin an exploration of how to route crowdsourcing task via Matrix factorization (MF) which efficiently estimate missing values in a worker-task matrix. Our preliminary results show the benefits of task routing over random assignment, the strength of probabilistic MF over baseline methods.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Selection process, Information filtering*

Keywords

crowdsourcing, quality assurance, matrix factorization, task routing, recommendation, collaborative filtering

1. PROBLEM

Crowdsourcing is quickly changing data collection practices in both industry and data-driven research areas such as natural language processing [16], computer vision [17], and information retrieval [1]. Despite its popularity in diverse areas, quality concerns persist, especially with rudimentary crowdsourcing platforms like Amazon's Mechanical Turk (MTurk) where factors such as anonymity, piece-work pay, and limited worker interaction can contribute to poor quality crowd work. Various statistical approaches have been proposed for mitigating these issues, such as by aggregating inputs from multiple workers [16] or filtering out inaccurate workers [13]. However, most studies have focused on how to reduce noise after collecting data from crowd workers. Little work to date has investigated effective matching of workers and tasks as another aspect of ensuring quality.

MTurk's default use case assumes workers self-select tasks. From a task requester's perspective, MTurk does not support any matching mechanism between a task and a worker. Moreover, lack of support for task routing in MTurk's default setup has led to a dearth of research in this area. Nonetheless, like spam filtering, the promise of work filtering and tailored work assignments is to better match workers to work for which they are best suited, with potential to increase work quality and satisfaction and reduce inefficiency of task selection.

We propose methods to route a crowdsourced task to a mostly appropriate worker for quality assurance. One of the naive way for routing is to simply predict a crowd worker's accuracy on new tasks based on his accuracy on past tasks. Such prediction provides a foundation for identifying the best workers to route work to in order to maximize accuracy on the new task. Note our methods do *not* require example feature representations and so are broadly applicable across crowdsourcing tasks. Our key insight, based on preliminary analysis on our MTurk data (Section 3.1), suggests cross-task worker accuracies being correlated based on task similarity. Intuitively, more similar tasks should yield more similar worker accuracy across tasks. Of course, "spammers" may still perform uncorrelated, inaccurate work across tasks. By modeling similarity to past tasks, work history can be better integrated to predict new task accuracy.

Critically, note that such prediction cannot be performed for one-shot or small scale data collection, where there is no work history, or where workers complete little work before leaving and never returning. While many academic studies have tended to report low rates of worker retention across tasks and within-task completion, we posit this may reflect community sampling bias of one-off, infrequent academic studies. In contrast, commercial crowd work offers large volume and repetition that allows workers to amortize time to spent learning a task, returning to tasks for which they are already familiar for greater retention and completion rates. While the MTurk data used here is relatively small (Section 3.1), it is drawn from such a use case where we see significant worker retention across tasks.

We describe two approaches to predict worker accuracies based on matrix factorization (MF), widely used in collaborative filtering problems to predict missing values in a matrix using low-rank feature vectors [10]. To predict unobserved workers' performance on a new task, we construct a worker-task matrix, where entries reflect a worker's observed accuracy on past tasks, evaluated against some sample of ground truth data (Figure 1). We investigate two well-known MF models: Singular Value Decomposition (SVD) and Probabilistic Matrix Factorization (PMF) [14]. Prior work [11] de-

WWW'14 Companion, April 7–11, 2014, Seoul, Korea.
ACM 978-1-4503-2745-9/14/04.

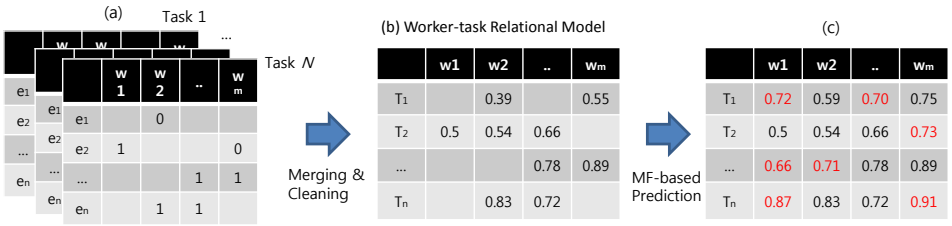

Figure 1: Matrix factorization (MF) prediction of crowd workers' accuracies using a worker-task matrix. (Left) A worker-example matrix contains labels from workers for each task. (Center) From the worker-example matrices, we measure each worker's accuracy vs. a ground truth sample, then merge all workers' accuracies into a single worker-task matrix. (Right) Unobserved workers' accuracies are predicted by MF (shown in red), and these predictions allow us to tailor individual work assignments to workers predicted to perform well for a given task.

scribes the two MF's tradeoffs for general recommendation systems. We revisit such questions by investigating these issues in our setting of crowd worker accuracy prediction.

Experiments on synthetic datasets provide feasibility assessment and comparative evaluation of MF approaches vs. three baseline methods. Across a range of data scales and task similarity conditions, we evaluate methods in terms of RMSE prediction error over all workers. The followings are principal research questions.

RQ1: MF-based Prediction Accuracy. *How does MF prediction performance vary as a function of task similarity, matrix size, and matrix density for predicting worker accuracies across tasks? How feasible and robust is it to challenging conditions?*

RQ2: Finding Top Workers. *Does task routing to predicted top k workers outperform random assignment? If so, by what degree, and how do proposed MF methods (PMF and SVD) compare vs. simpler baseline methods (average and weighted average)?*

RQ3: The Effect of Spammers. *How robust is MF-based task routing to the existence of spammers?*

2. STATE OF THE ART

MTurk's standard method of task self-selection has led to relatively few studies on task routing to better match workers to tasks, though work considered task assignment in other venues, such as Wikipedia [4]. Others have studied the cooperative refinement and task routing among online agents with regard to prediction tasks [7]. Bernstein at al. [2] investigate task routing in terms of real-time crowdsourcing. Though informative, these studies do not address finding strong candidates for a particular task from a task requester's viewpoint.

Karger et al. [8] present a task assignment model based on random graph generation and a message-passing inference algorithm, in order to route tasks to crowd workers under homogeneous labeling tasks. Ho et al. [5] attempt to generalize this model to allow heterogeneous tasks by applying on-line primal-dual techniques. However, neither study answers the question of how to predict the unobserved workers' performance, which is critical to task routing in practice. Zhang et al. [20] consider a related task routing task that seeks to engage people or automated agents to both contribute solutions and route tasks onward.

Jung and Lease [6] study MF methods to improve the quality of crowdsourced labels, using a PMF model to infer unobserved labels in order to reduce the bias of the existing

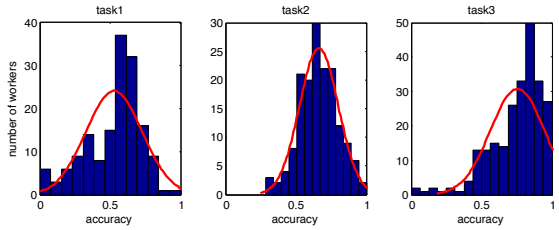

Figure 2: Histograms of Workers' Accuracy in three different tasks. All these three histograms show that workers' accuracy distribution follow a normal distribution.

labels. They do not consider task routing. Yi et al. investigate matrix completion for crowdclustering, and more recently inferring user preferences [18]. Yuen et al. [19] consider workers' task selection preferences and propose a task recommendation model based on PMF. However, they motivated their approach on conceptual grounds and did not evaluate it. Most recently, Kolobov et al. [9] investigate task routing of multiple tasks across a common pool of workers.

3. PROPOSED APPROACH AND METHODOLOGY

3.1 Crowdsourced Data Analysis

In this study, we first use synthetic data, letting us carefully control a variety of experimental variables for detailed analysis. Following this, we plan to do additional experiments with real crowd data to assess performance of methods for a specific case of actual operating conditions.

Prior to describing how to generate synthetic data, we first investigate the accumulated real MTurk dataset consisting of three tasks. Figure 2 plots histograms showing the number of workers achieving various levels of accuracy in each of three MTurk tasks. These histograms show a strong normal tendency, which we quantify later via a Shapiro-Wilk test [15] (Table 1). Besides, Figure 3 (left) plots average worker accuracy for task 1 vs. task2 and shows strong correlation at high accuracies (similar plots for task 1 vs. task 3 and task 2 vs. task 3 are not shown). In other words, the best workers appear to be fairly accurate across tasks, whereas other workers show less correlation across tasks, perhaps tending toward uncorrelated accuracies as average accuracy across tasks decreases (e.g., the increasing prevalence of "spammers" as we consider lower average accuracies). These observations suggest correlated worker accuracies across tasks, which might be reasonably well-fit by

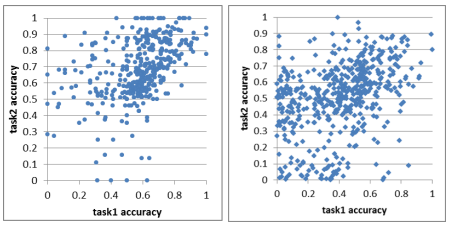

Figure 3: Scatter plots of accuracies across two tasks in (a) MTurk data and (b) synthetic data.

a multivariate normal distribution with appropriate covariance. We further develop this idea below.

3.2 Data Preparation

Our model of worker behavior makes three key assumptions. Firstly, we model workers' accuracies as following a normal distribution N(μ, σ), based on our real crowd accuracy histograms (Figure 2) and supported by a Shapiro-Wilk test [15] confirming high normality (Table 1). Secondly, we assume worker accuracy is correlated across tasks in proportion to task similarity. In addition to knowing people naturally exhibit varying expertise/skill across tasks, we also observe this in our real crowd data (correlation in Figure 3's left plot). Thirdly, we posit the existence of "spammers" who exhibit low accuracy, uncorrelated across tasks, due to any number of factors, such as fatigue, low language competency, negligence, etc. Because better workers are less likely to be spammers (by definition), we expect fewer spammers when average worker accuracy is high, and the ratio increasingly shifting toward more spammers at lower accuracies.

Algorithm 1 Generative model for synthetic data

Input: $taskSimilarity, numWorkers, numTasks$
$\sigma = matrix(taskSimilarity, numWorkers, numTasks)$
$m = multivarNormal(numWorkers, \mu = 0.5, \sigma)$
for each $t \in [1 : numTasks]$ **do**
  $accuracy[1 : numWorkers] = m[t]$
  for each $j \in [0 : 0.8]$ by 0.1 **do**
    $strata = [j, j + 0.1]$
    $workers = findWorkers(accuracy, strata)$
    $\alpha = (0.8 - j) * 10$ ▷ $\alpha \in \{80, 70, \ldots, 0\}$
    $subset = getRandomSample(workers, \alpha\%)$
    for each $worker \in subset$ **do**
      max =0.9/(9-j*10) ▷ $max \in \frac{0.9}{[9,8,7,\ldots,1]}$
      $accuracy[worker] = uniform[0, max]$
Output: m

Our generative model for synthetic data (Algorithm 1) takes three parameters as input which we vary as experimental variables: task similarity s, number of tasks t, and number of workers w. We sample a multivariate normal distribution of accuracies (per-task average accuracy), providing correlation across tasks by specifying covariance matrix σ filled with s (`rmvnorm` function from `mvtnorm` library in R). This "optimistic" distribution is free of spammers and reflects an idealized model of correlated accuracies. Next, we introduce spammers to produce an alternative "pessimistic" distribution by transforming a percentage of the idealized workers into spammers. To accomplish this, we first group workers by distributional strata over average accuracy across tasks (using a sliding window of size 0.1). We then randomly sample a percentage of diligent workers from each strata and transform them into spammers by replacing their idealized per-task accuracy on each task with an accuracy sampled uniformly at random.

Tasks	Workers	Accuracy	Normality Test
Task1	206	0.676	0.9471 *
Task2	384	0.599	0.99 *
Task3	167	0.491	0.90 *
All	443	0.596	

Table 1: Attributes of MTurk data with three tasks. Task similarity s ranges from [0.545:0.719]. For all tasks, distribution of per-task worker accuracies follow a normal distribution with high statistical significance ($p < 0.01$) under the Shapiro-Wilk test [15].

Note that as a function of strata, we must decide (a) what percentage of workers to transform, and (b) the maximum accuracy of the interval from which to sample spammer accuracies. While our choice of functions here for (a) and (b) are relatively ad hoc, they enforce our principle above of finding fewer spammers as worker accuracy increases. We argue strengths of this model include: (i) its full description for reproducibility by others [12]; (ii) its implementation of over-arching modeling assumptions in some reasonable way; and (iii) an explicit discussion of how more accurate simulation might be achieved by further analysis and characterization of real crowd data properties.

3.3 Worker-Task Matrix Factorization

Matrix factorization (MF) has been studied for effectively recommending an item for a user in an online marketplace and advertisement. Our intuition is that finding strong candidates for a specific task is very similar to the recommendation of items for a specific user in collaborative filtering [6]. In addition, latent features should capture how a worker successfully makes a label for a specific task. For example, two workers would achieve high accuracy for a task type if they both have similar amounts of domain knowledge for this task type. If we can discover these latent features, we should be able to predict workers' accuracies by task. Given a partially observed worker-task matrix, we aim to predict unobserved workers' accuracies (e.g., on new tasks) so that we might route work optimally, or recruit or exclude particular workers for a given new task.

3.3.1 Singular Value Decomposition

Singular Value Decomposition (SVD) seeks an approximation matrix $\hat{R} = W^T T$ of the given rank which minimizes the sum-squared distance between the original matrix R and $\hat{R}$. It has a critical drawback: it is dependent only on the observed elements in R and is undefined for missing elements. Thus, it does not handle the problem of sparseness in the given R.

3.3.2 Probabilistic Matrix Factorization

Probabilistic Matrix Factorization (PMF) was introduced by Ruslan [14] and has demonstrated excellent performance in the Netflix challenge. We have M crowd workers, N tasks, and a worker-task matrix R in which R_{ij} indicates the accuracy of worker i for task j. Let $W \in \mathbb{R}^{D*M}$ and $T \in \mathbb{R}^{D*N}$ be latent feature matrices for workers and tasks, with column vectors W_i and T_j representing D-dimensional crowd worker-specific and task-specific latent feature vectors, re-

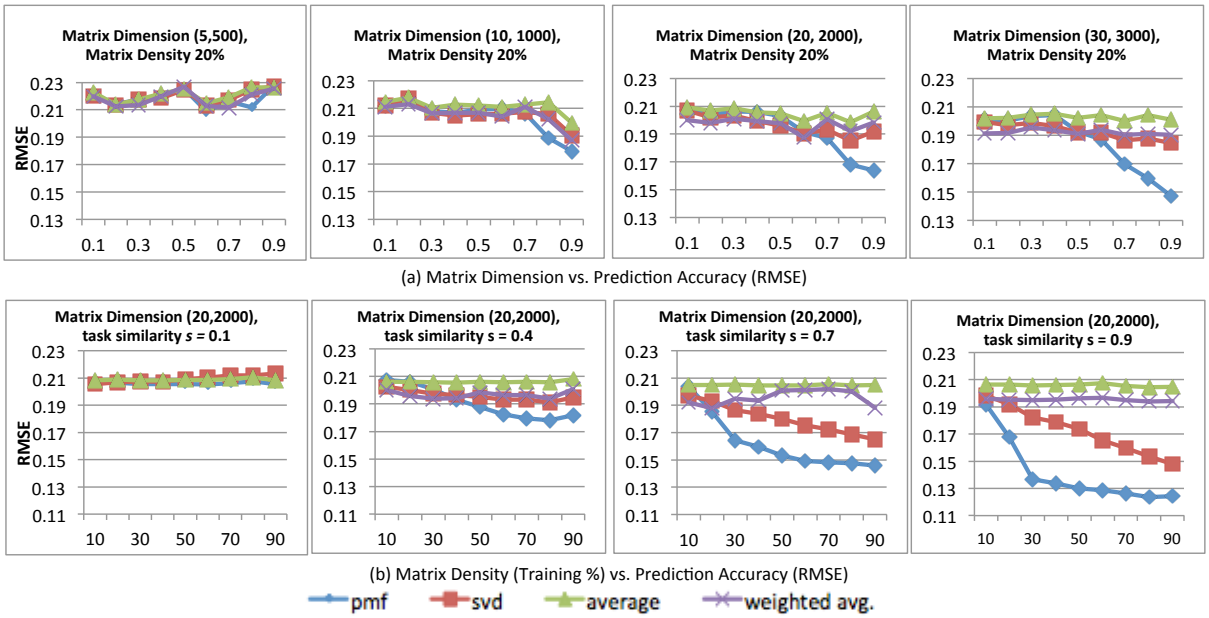

Figure 4: (a) Matrix dimension vs. prediction accuracy and (b) Matrix Density (Training %) vs. prediction accuracy. In (a), the x-axis shows the increase of task similarity ranging from 0.1 to 0.9 by 0.1. In order to investigate the effect of matrix dimension, we evaluate the prediction accuracy by increasing the number of tasks (t) and the dimensionality of feature vectors ($d = t-1$). Matrix density is fixed as 20%. In (b), the x-axis shows the increase of matrix density from 10% to 90% by 10%. In addition, we increase task similarity from 0.1 to 0.9 while fixing matrix dimension as 20 by 2,000 and the dimensionality of feature vectors ($d = 19$).

spectively.Indicator I_{ij} equals 1 *iff* worker i's accuracy is measured over task j. We place zero-mean spherical Gaussian priors on worker and task feature vectors. To estimate model parameters, we maximize the log-posterior over example and worker features with fixed hyper-parameters. Maximizing the posterior with respect to W and T is equivalent to minimizing squared error with L2 regularization:

$$E = \frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{N} I_{ij}(R_{ij} - W_i^T T_j)^2 + \frac{\lambda_W}{2}\sum_{i=1}^{M}\|W_i\|_{Fro}^2 + \frac{\lambda_T}{2}\sum_{j=1}^{N}\|T_j\|_{Fro}^2 \quad (1)$$

where $\lambda_W = \sigma_W/\sigma, \lambda_T = \sigma_T/\sigma$, and $\|\cdot\|_{Fro}^2$ denotes the Frobenius Norm. We use gradient descent to find a local minimum of the objective for W and T. Finally, we infer unobserved workers' accuracies in the worker-task matrix R by the scalar product of W and T.

3.4 Experiment Setting

Metrics. We report root-mean squared error (RMSE):

$$RMSE = \sqrt{\frac{\sum_{i,j}(\hat{a}_{i,j} - a_{i,j})^2}{n}}$$

where $\hat{a}_{i,j}$ and $a_{i,j}$ respectively denote the predicted vs. observed worker accuracy for worker i on task j.

While task similarity s could be operationalized in many ways, in this study we adopt Pearson's correlation coefficient r (i.e., $s = r$ in our study, but we preserve distinct notation of s for generality). Following standard practice [3], we distinguish four specific levels of correlation in our experiments: weak (r=0.1), medium (r=0.4), strong (r=0.7), and very strong (r=0.9).

PMF model parameters. In order to find PMF regularization parameters λ_w and λ_t, learning rate ϵ, an optimal dimensionality D, we do 5-fold cross-validation, using **only** 20% of data for training and leaving 80% for testing. From the process, we ultimately select $\lambda_w = \lambda_t = 0.01$ and D=$t-1$ (t is the number of tasks). We use $\epsilon = 0.005$ without tuning.

Baselines One goal of our work is to assess the potential benefit of task routing at all (i.e., any approach) vs. just assigning tasks arbitrarily (i.e., random task assignment). For comparative experiments, we use three types of baselines: a simple random assignment (MTurk), average, and weighted average. The second baseline is to simply predict a worker's accuracy on a new task by his average accuracy on other tasks. A limitation of this baseline, however, is that it completely ignores task similarity. Moreover, since our generative model for synthetic data explicitly generates worker accuracies correlated across tasks by similarity (Section 3.2), we expect a baseline exploiting this additional information should perform better. Our third baseline therefore computes an expectation (weighted average) across tasks based on task similarity rather than a simple average.

4. RESULTS

To what extent do task similarity, matrix size, and matrix density influence MF-based prediction of crowd worker accuracies for varying task similarity? We first measure RMSE of PMF and SVD methods vs. baseline methods for task similarity s (Pearson's correlation r) $\in \{0.1, 0.2, \ldots, 0.9\}$.

Our first set of experiments (Figure 4(a)) evaluate our ability to effectively predict worker accuracies across varying task similarity s, number of tasks, and number of workers. We vary the number of tasks from 5 to 30 (by 5) and the number of workers from 500 to 3000 (by 500). Task similarity s is varied along the x-axis. Matrix density is fixed

at 20% and dimensionality D is set by $D = t - 1$. The second set of experiments (Figure 4(b)) evaluates our ability to effectively predict worker accuracies under varying task similarity s and matrix density.

Task Similarity. With only weak similarity $s = 0.1$ (left-most point in all 4 plots in (a), left-most plot in (b)), RMSE of 0.21-0.23 can still be achieved, though this is far below the best RMSE<0.13 observed with very high task similarity. Baselines seem sufficient, without need for MF methods. As task similarity increases (across x-axis in (a) plots, and across plots in (b)), it tends to enable better prediction of worker accuracies across tasks, as expected. Moreover, we see PMF predictions improve by exploiting greater task similarity: both as the number of tasks and workers increase (a), or as matrix density increases (b). In contrast, SVD performs comparably to baselines in all four of the plots in (a); it does not benefit from increased task similarity even as the matrix size grows. While in (b) plots we do see SVD perform much better as s increases with greater density, PMF still outperforms SVM by a wide margin.

Matrix Size. Figure 4(a) shows that in the smallest case of 5 tasks and 500 workers, RMSE of 0.21-0.23 can still be achieved. Moreover, baseline methods seem sufficient in this case (comparable to MF methods). However, this is far below the sub-0.15 RMSE achieved by PMF with larger matrix sizes. As noted above, we see PMF capitalize on increasing task similarity s with larger matrices while SVD does not.

The Effect of Matrix Density/Sparseness. Prior crowdsourcing studies often report workers are often transient and rarely complete all tasks available. For this concern, this experiment gives an answer. Figure 4(b) shows across plots, RMSE lies in [0.19-0.21] when density is only 10%. As before, baseline methods seem sufficient when density is so low, with no improvement from MF methods. However, as noted above, both SVD and PMF improves dramatically vs. baselines with greater density, with PMF consistently dominating SVD across (b) plots. The best case RMSE<0.13 is observed under ideal settings of an extremely dense matrix and large worker-task matrix.

RQ1 Summary. Under worst case conditions, results show worker accuracies can be predicted with RMSE [0.19-0.23], and that even simple averaging suffices is comparable to MF methods. However, as tasks exhibit even medium task similarity ($s = 0.4$) significant gains are possible, and especially with steep with high correlation ($s = 0.7$). Similar wins occur as matrix size increases, and with steep RMSE decreases beginning with matrix densities around only 30%. PMF consistently outperforms SVD in all cases where sufficient data exists to outperform baselines.

5. CONCLUSIONS AND FUTURE WORK

Task routing represents a relatively little explored and increasingly important area for future quality improvements in crowdsourcing. In this stage, we are still investigating the proposed method and looking for the answers of RQ2 and RQ3. In addition, we plan to verify the feasibility of the proposed method in practice by collecting large-scale MTurk data for this research.

Many interesting questions remain open, such as modeling time-varying worker accuracies, due to fatigue or training effects, in determining appropriate task routing. Better ways to tackle sparse worker history data and longer term longitudinal worker studies could also be incredibly informative. Consequently, task routing in crowdsourcing could bring substantial benefits in terms of quality assurance.

6. REFERENCES

[1] Alonso, O., and Baeza-Yates, R. Design and implementation of relevance assessments using crowdsourcing. In *Proceedings of the ECIR'11* (Berlin, Heidelberg, 2011), pp. 153–164.

[2] Bernstein, M. S., Karger, D. R., Miller, R. C., and Brandt, J. Analytic methods for optimizing realtime crowdsourcing. In *Collective Intelligence* (2012).

[3] Cohen, J. *Statistical power analysis for the behavioral sciences*, 2nd ed. Lawrence Erlbaum Assoc., 1988.

[4] Cosley, D., Frankowski, D., Terveen, L., and Riedl, J. Suggestbot: using intelligent task routing to help people find work in wikipedia. In *In Proceedings of IUI'07* (2007), pp. 32–41.

[5] Ho, C., Jabbari, S., and Vaughan, J. W. Adaptive task assignment for crowdsourced classification. In *Proceedings of the ICML'13* (2013), pp. 534–542.

[6] Jung, H. J., and Lease, M. Improving Quality of Crowdsourced Labels via Probabilistic Matrix Factorization. In *Proceedings of the AAAI HCOMP workshop'12* (2012), pp. 101–106.

[7] Kamar, E., Hacker, S., and Horovitz, E. Combining human and machine intelligence in large-scale crowdsourcing. In *Proceedings of the AAMAS'12* (2012), pp. 467–474.

[8] Karger, D. R., Oh, S., and Shah, D. Iterative learning for reliable crowdsourcing systems. In *Proceedings of the NIPS'11* (2011), pp. 1953–1961.

[9] Kolobov, A., Mausam, and Weld, D. S. Joint Crowdsourcing of Multiple Tasks. In *Proceedings of the HCOMP'13)* (2013).

[10] Koren, Y., Bell, R., and Volinsky, C. Matrix factorization techniques for recommender systems. *IEEE Computer 42*, 8 (2009), 30–37.

[11] Lee, J., Sun, M., and Lebanon, G. A comparative study of collaborative filtering algorithms. *CoRR abs/1205.3193* (2012).

[12] Paritosh, P. Human computation must be reproducible. In *CrowdSearch: WWW Workshop on Crowdsourcing Web Search* (2012), pp. 20–25.

[13] Ramesh, A., Parameswaran, A., Garcia-Molina, H., and Polyzotis, N. Identifying reliable workers swiftly. Tech. rep., Stanford InfoLab, 2012.

[14] Salakhutdinov, R., and Mnih, A. Probabilistic matrix factorization. In *Proceedings of NIPS'08* (2008).

[15] Shapiro, S., and Wilk, M. An analysis of variance test for normality (complete samples). *Biometrika 3*, 52 (1965).

[16] Snow, R., O'Connor, B., Jurafsky, D., and Ng, A. Cheap and fast – but is it good?: evaluating non-expert annotations for natural language tasks. In *Proceedings of the EMNLP'08* (2008), pp. 254–263.

[17] Vijayanarasimhan, S., and Grauman, K. Large-scale live active learning: Training object detectors with crawled data and crowds. In *Proceedings of IEEE CVPR'11* (2011).

[18] Yi, J., Jin, R., Jain, A. K., and Jain, S. Crowdclustering with sparse pairwise labels: A matrix completion approach. In *Proceedings of the AAAI HComp workshop* (2012), pp. 47–53.

[19] Yuen, M., King, I., and Leung, K.-S. Task recommendation in crowdsourcing systems. In *Proceedings of the CrowdKDD workshop* (2012), pp. 22–26.

[20] Zhang, H., Horvitz, E., Chen, Y., and Parkes, D. C. Task routing for prediction tasks. In *Proceedings of the AAMAS'12* (2012), pp. 889–896.