Human Computation
AAAI Technical Report WS-12-08

Improving Quality of Crowdsourced Labels via Probabilistic Matrix Factorization

Hyun Joon Jung
School of Information
University of Texas at Austin
hyunJoon@utexas.edu

Matthew Lease
School of Information
University of Texas at Austin
ml@ischool.utexas.edu

Abstract

Quality assurance in crowdsourced annotation often involves having a given example labeled multiple times by different workers, then aggregating these labels. Unfortunately, the worker-example label matrix is typically sparse and imbalanced for two reasons: 1) the average crowd worker judges few examples; and 2) few labels are typically collected per example to reduce cost. To address this missing data problem, we propose use of probabilistic matrix factorization (PMF), a standard approach in collaborative filtering. To evaluate our approach, we measure accuracy of consensus labels computed from the input sparse matrix vs. the PMF-inferred complete matrix. We consider both unsupervised and supervised settings. In the supervised case, we evaluate both weighted voting and worker selection. Experiments are performed on both a synthetic data set and a real data set: crowd relevance judgments taken from the 2010 NIST TREC Relevance Feedback Track.

Introduction

Crowdsourced labeling offers potential to reduce time, cost, and effort of obtaining relevance judgments used to evaluate search engine ranking algorithms (Alonso, Rose, and Stewart 2008). However, quality of judgments from non-workers continues to be a concern, motivating continuing work in quality assurance methods based on statistical label aggregation methods or greater attention to human factors. A common approach is to collect multiple, redundant judgments from workers and aggregate them via methods like majority voting (MV) or expectation maximization (EM) to produce consensus labels (Ipeirotis, Provost, and Wang 2010).

Unfortunately, the average crowd worker typically judges only a small number of examples. Moreover, few labels are typically collected per example to reduce cost. As a result, collected judgments are typically sparse and imbalanced, with the consensus judgment for each example determined by only a handful of workers. MV is completely susceptible to this problem. EM addresses this indirectly: while only workers labeling an example vote on it, global judgments are used to infer class priors and worker confusion matrices.

We propose to tackle this issue via a collaborative filtering approach, a popular strategy to address sparsity of user ratings (e.g., movies, books, etc.). In particular, we employ probabilistic matrix factorization (PMF), which induces a latent feature vector for each person and example (Salakhutdinov and Mnih 2008) in order to infer unobserved judgments for all examples. Figure 1 depicts our approach. PMF exploits latent feature matrices of workers and examples, with gradient descent used to find a local minimum of the objective for the worker and example feature vectors. Inference yields a complete matrix, which we then use use for label aggregation. This complete matrix contains relevance judgments from all workers corresponding to all examples, thereby reducing the bias of output consensus labels.

To evaluate our PMF approach, we measure accuracy of consensus labels computed from the input sparse matrix vs. the PMF-inferred complete matrix. We consider both unsupervised and supervised settings. In the supervised case, we evaluate both weighted voting and worker selection. Experiments are performed on both a synthetic data set and a real data set: crowd judgments collected from Amazon Mechanical Turk for the 2010 NIST TREC Relevance Feedback Track (Buckley, Lease, and Smucker 2010). We do not know of prior work investigating PMF or other collaborative filtering approaches for quality assurance in crowdsourcing.

The rest of this paper is organized as follows. The next section summarizes prior work on PMF methods and quality issues in crowdsourcing. Following this, we present algorithmic background of PMF in the crowdsourcing context. Next, we describe label aggregation methods in unsupervised and supervised cases, with the latter considering weighted voting & filtering methods. We then describe the synthetic data set and the real data used in our experiments. Next, evaluation results compare performance between the proposed method and the other aggregation methods. We conclude the paper with discussion and future work.

Related Work

Label acquisition varies according to the number of labels per example (single vs. multiple) and the number of examples per worker (single expert vs. multiple crowd workers). One naive way is that one expert labels all examples. It would be expertise based label acquisition rather than crowd-worker based method since labels from crowd work-

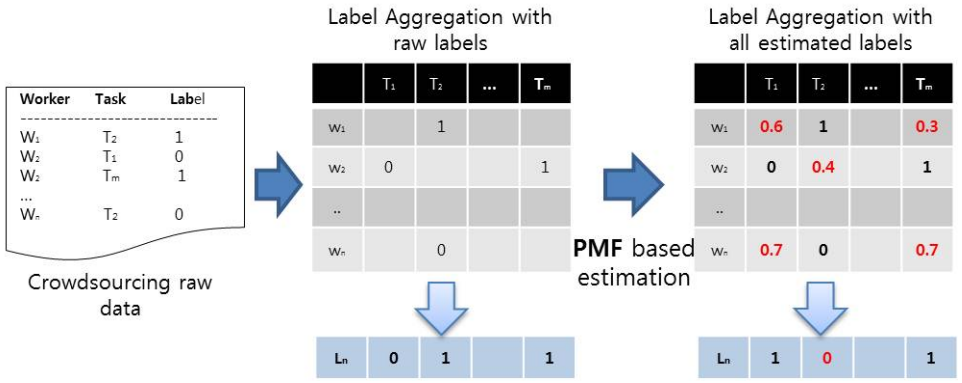

Figure 1: Crowdsourcing workers judgments (Left) are copied to a sparse worker-example matrix (Middle). Missing judgments are inferred via PMF (Right). L_n is a set of consensus labels correpsonding to each task induced by a label aggregation method.

ers may be not as reliable as expert does due to the lack of knowledge and experience. However, this approach leads to very expensive cost and comparatively long time. Another way is to distribute all examples into multiple crowd workers by assuming one label per each example (single labeling). This method saves cost and time by using the wisdom of crowds; however, the quality of label is highly dependent on single worker's subjectivity and knowledge.

To solve the limit of single labeling methods, multiple labeling approaches based on redundancy have been proposed as an efficient way to integrate the labels from many annotators (Sheng, Provost, and Ipeirotis 2008) (Welinder and Perona 2010). When multiple labels are available, a critical issue is how to aggregate labels efficiently and accurately. One simple way is to induce consensus labels based on majority voting. However, it is nothing more than random label selection when the disagreement between multiple labels goes high. Thus, many studies have focused on improving the accuracy of consensus labels by predictive models with ground truth (Snow et al. 2008) (Dekel and Shamir 2009) or without ground truth (Ipeirotis, Provost, and Wang 2010).

Dawid et al. (Dawid and Skene 1979) presented a model for multi-value annotations where the biases and skills of the annotators were modeled by a confusion matrix. Welinder and Perona generalized and extended it to different annotation types (Welinder and Perona 2010). In a similar way, Raykar et al.(Raykar et al. 2010) presented a model that considered annotator bias in the context of training binary classifiers with noisy labels. Whitehill et al. (Whitehill et al. 2009) modeled both annotator competence and example difficulty, while did not consider annotator bias.

All these approaches are based on given raw labels which were actually annotated. When all workers annotate the same number of examples, worker's quality such as accuracy over gold is measured over same number of labels and each worker influences consensus labels uniformly. However, each worker may annotate a different number of examples unless a requester limits it. In reality, this issue occurs more seriously as shown in Figure 2. Thus, we attempt to deal with this issue by focusing on how to reduce the bias of given labels with PMF.

To the best of our knowledge, we are not familiar with any prior work investigating PMF, or collaborative filtering approaches more generally, toward crowdsourcing quality assurance. Related prior work has investigated other ways to infer bias corrected labels in place of raw labels (Ipeirotis, Provost, and Wang 2010), as well as inference of missing labels by estimating a unique classifier for each worker (Chen et al. 2010).

Probabilistic Matrix Factorization (PMF)

Suppose we have *M* examples, *N* workers, and a label matrix R in which R_{ij} indicates the label of worker i for example j. Let $U \in \mathbb{R}^{D*M}$ and $V \in \mathbb{R}^{D*N}$ be latent feature matrices for workers and examples, with column vectors U_i and V_j representing D-dimensional worker-specific and example-specific latent feature vectors, respectively. The conditional probability distribution over the observed labels $R \in \mathbb{R}^{N*M}$ is given by Equation 1. Indicator I_{ij} equals 1 *iff* worker i labeled example j. We place zero-mean spherical Gaussian priors on worker and example feature vectors (Equations 2 and 3).

$$p(R|U,V,\sigma^2) = \prod_{i=1}^{N}\prod_{j=1}^{M}[\mathcal{N}(R_{ij}|U_i^T V_j, \sigma^2)]^{I_{ij}} \quad (1)$$

$$p(U|\sigma_U^2) = \prod_{i=1}^{N}[\mathcal{N}(U_i|0, \sigma_U^2 \mathrm{I})] \quad (2)$$

$$p(V|\sigma_V^2) = \prod_{j=1}^{M}[\mathcal{N}(V_j|0, \sigma_V^2 \mathrm{I})] \quad (3)$$

To estimate model parameters, we maximize the log-posterior over example and worker features with fixed hyper-parameters. Maximizing the posterior with respect to U and V is equivalent to minimizing squared error with L2

regularization:

$$E = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{M} I_{ij}(R_{ij} - U_i^T V_j)^2$$

$$+\frac{\lambda_U}{2}\sum_{i=1}^{N}\|U_i\|_{Fro}^2 + \frac{\lambda_V}{2}\sum_{i=1}^{M}\|V_j\|_{Fro}^2$$

where $\lambda_U = \sigma_U/\sigma, \lambda_V = \sigma_V/\sigma$, and $\| \cdot \|_{Fro}^2$ denotes the Frobenius Norm. We use gradient descent to find a local minimum of the objective for U and V. Finally, we infer missing worker judgments in the worker-example matrix T by taking the scalar product of U and V. Note that as in (Ipeirotis, Provost, and Wang 2010), we also replace actual labels with bias-corrected inferred labels.

Label Aggregation

Label aggregation is to generate a consensus label per each example with a set of inferred relevance judgments labels. In order to compare the effectiveness of the proposed method, we consider several ways to aggregate labels with or without supervision. We first introduce unsupervised methods to aggregate labels. We subsequently compare the proposed method with supervised ways by using training gold labels.

Unsupervised Label Aggregation

Given the complete set of inferred worker relevance judgments in matrix R, we next aggregate worker judgments to induce consensus labels. Majority Voting is a straightforward method which takes account of each label uniformly. We consider majority voting with raw (sparse) labels as a baseline (Method 1).

Expectation Maximization (EM) (Dawid and Skene 1979) estimates the error rates of each classifier c_k by a latent *confusion matrix* $[\pi_{ij}^{(k)}]$, where ij-th element $\pi_{ij}^{(k)}$ denotes the probability of classifier c_k classifying an example to class j given the true label is i, estimated based on each example's class membership as:

$$\hat{\pi}_{ij}^{(k)} = \frac{\sum_{m=1}^{M} T_{mi} n_{mj}^{(k)}}{\sum_{i=1}^{C}\sum_{m=1}^{M} T_{mi} n_{mj}^{(k)}} \quad (4)$$

where $n_{mj}^{(k)}$ denotes the number of times label l_m receives response j from classifier c_k, and $\{T_m i\}$ represents the set of indicators for class membership of label l_m such that $T_{mt} = 1$ if t is the true label for label l_m and T_{mi} = 0 otherwise. The latent class prior $\{p_i\}_{i=1}^{L}$ is estimated by:

$$\hat{p}_i = \frac{1}{M}\sum_{m=1}^{M} T_{mi} \quad (5)$$

Since the true label for each label l_m is unknown in the unsupervised methods, EM uses a mixture of multinomials to describe the quality of classifiers. Assuming every pair of classifiers is independent, the probabilistic model likelihood can be written:

$$L(p_i, \pi_{ij}^{(k)}) = \prod_{m=1}^{M}\left(\sum_{i=1}^{C} p_i \sum_{k=1}^{K}\sum_{j=1}^{C}(\pi_{ij}^{(k)})^{n_{mj}^{(k)}}\right) \quad (6)$$

Estimating the maximum likelihood in Equation 6 is analytically intractable since it involves computing the product of a summation. However, once we get estimates for latent parameters p_i and $\pi_{ij}^{(k)}$, we can derive new class membership $T_m i$ for label l_m such that $T_m l$ = 1 if l becomes the estimated true label for label l_m which maximizes:

$$L(p_i, \pi_{ij}^{(k)}) = \prod_{m=1}^{M} p_i \prod_{k=1}^{K}\prod_{j=1}^{C}(\pi_{ij}^{(k)})^{n_{mj}^{(k)}}. \quad (7)$$

We then iteratively re-estimate latent p_i and $\pi_{ij}^{(k)}$, and missing labels T_{mi} from Equations 4, 5, and 7 until convergence. We use this method with raw labels as another baseline (Method 2).

Supervised Label Aggregation

In supervised setting, we measure each worker's accuracy based on expert judgments. We flip labels of anti-correlated workers in order to make accuracy always $\geq 50\%$. We use supervision in two distinct ways: weighted voting (WV) and worker filtering. For weighted voting, we put each worker's accuracy as an weight on each label. For worker filtering, only workers with accuracy $\geq \alpha$ participate in voting. As shown in Table 1, method 4 uses only weighted voting, and method 5 is only based on worker filtering method. Method 6 exploits both methods by filtering out noisy labels then doing weighted voting.

Data

Synthetic Data

We generate a synthetic data set for binary classification with 10,000 examples (uniformly) randomly assigned to each class. We generate a pool of 1000 workers. Each worker's accuracy is randomly selected between 0.2 and 0.8, and the distribution of worker's accuracy follows a normal distribution of $N(0.5, 1)$. The number of examples per worker follows an exponential probabilistic distribution with a parameter μ=40. No examples are annotated by the same worker more than one time. Since the size of the given worker by example matrix is 10,000,000 and the number of generated labels is approximately 40,000, only 4/1000 = 0.4% of labels in the raw matrix is given. It is similar to the case we face in Turk data which will be described in the next section.

Turk Data

Turk data contains crowd judgments collected in the 2010 TREC Relevance Feedback Track (Buckley, Lease, and Smucker 2010) from Amazon Mechanical Turk. 762 crowd workers judged 19033 query-document examples, and 89624 judgments were collected. Our worker-example matrix thus has 762 columns (workers) and 19,033 rows (examples); only 89,624 out of 14,503,146 labels (0.6%) are observed, so data is extremely sparse. We use two sets of ground truth. First, we use 3,275 expert relevance judgments by NIST that are partitioned into training (2,275) and test (1,000) sets. The test set is evenly-balanced between relevant and non-relevant classes. Second,we use 1,865 *in-house*

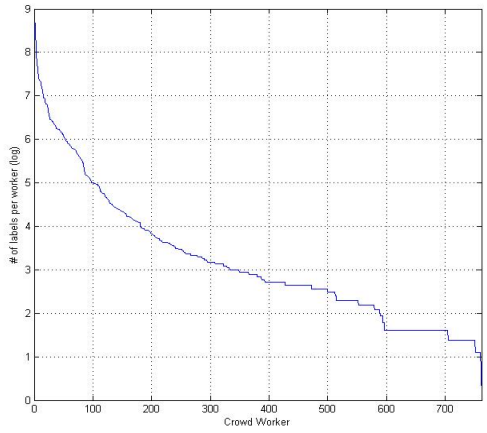

Figure 2: The number (log-scale) of examples per worker on crowd judgments collected in the TREC Mechanical Turk data set used.

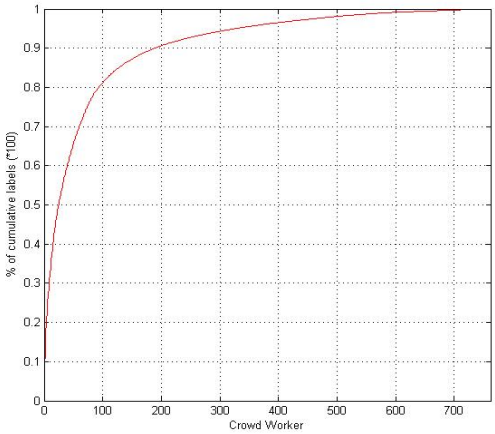

Figure 3: The percentage of cumulative labels per worker on crowd judgments collected in the TREC Mechanical Turk data set used.

relevance judgments produced by the University of Waterloo (Smucker 2012).

Figure 2 shows the number of examples per worker on crowd judgments collected in our turk data. In this data, requester did not limit the number of examples per worker, thus the distribution of the number of examples per worker is very imbalanced. Furthermore, the top 200 workers did label 90% of the given examples as shown in Figure 3. In this case, a crowd worker may have a serious effect on the quality of consensus labels more than the others since the number of examples per worker is skewed. Thus, a worker-by-example table is very sparse and imbalanced and it is very similar to a problem in collaborative filtering. Multiple labeling methods enable us to reduced noise of labels; however, it still suffers sparsity and imbalance since a crowdworker usually participates a small number of examples.

Evaluation

Evaluation Setting

Since PMF is fully-unsupervised, we iteratively run PMF with the entire set of raw labels to optimize parameters. For dimensionality of worker and example latent feature vectors, we consider $D \in 10, 30, 50$ and select $D = 30$ based on cross-validation on the entire set of labels (unsupervised) for both synthetic and turk data. We similarly tune regularization parameter $\lambda \in \{0.001, 0.01, 0.1, 0.3, 0.5\}$ and select $\lambda = 0.3$ for a synthetic data and $\lambda = 0.1$ for a turk data.

For a synthetic experiment, we compare the performance of PMF based inference method with majority voting (MV) and Expectation Maximization (EM). These are all unsupervised label aggregation methods that do not assume the presence of ground truth.

With a turk data, we conduct experiments with various label aggregation methods in unsupervised and supervised setting. In unsupervised setting, PMF based inference are evaluated with MV (method 1) and EM (method2). For supervised label aggregation, each worker's accuracy are used to filter out noisy workers and weighted voting. We tune the worker filtering threshold $\alpha \in [0.6, 0.99]$ by cross-validation on the training set using a linear sweep with step-size 0.01. We used the following measures for comparing the performance of given methods.

$$RMSE = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(p-q)^2} \tag{8}$$

$$Accuracy(ACC) = \frac{tp+tn}{(tp+tn+fp+fn)} \tag{9}$$

$$Specificity(SPE) = \frac{tn}{(tn+fp)} \tag{10}$$

where tp is the number of true positive classifications, fp is false positives, tn is true negatives, and fn is false negatives. p is the true labels and q is the predicted labels. $RMSE$ is a largely used measure of the difference between predicted values and actual values, which is popular in the evaluation of collaborative filtering algorithms. $Accuracy$ reports the proportion of true results over all inference labels, while $specificity$ indicates the ability of method to identify negative results. Thus, a specificy 100% indicates that the inference method recognizes all actual negatives correctly.

Results

Synthetic data Figure 4 shows the results of each methods in the absence of supervision. In terms of accuracy and RMSE, EM with raw labels and PMF based inference achieve the equivalent scores. A two-tailed paired t-test proves that there is no statistically significant difference between two methods at the significance level of 0.05. PMF reports a higher specificity score compared to two other methods, since it reduces the number of false positives and in-

Method	Supervised	Worker Labels	Label Aggregation	ACC	Rank	RMSE	Rank	SPE	Rank
1	No	raw (sparse)	MV	0.603	4	0.630	4	0.332	6
2	No	raw (sparse)	EM	**0.644**	**3**	**0.596**	**3**	0.418	5
3	No	PMF (complete)	MV	**0.643**	**3**	**0.598**	**3**	**0.440**	4
4	Yes	raw (sparse)	WV	0.642	3	0.598	3	**0.900**	**1**
5	Yes	raw (sparse)	Filtering(α=0.67)	**0.752**	**1**	**0.498**	**1**	0.838	2
6	Yes	raw (sparse)	WV & Filtering(α=0.67)	**0.750**	**1**	**0.500**	**1**	0.848	2
7	Yes	PMF (complete)	WV & Filtering(α=0.7)	0.673	2	0.571	2	0.542	3

Table 1: Results of PMF-based inference of missing worker labels over NIST ground truth. For the unsupervised case, majority voting (MV) with PMF (Method 3) is compared to MV and EM approaches using input (sparse) worker labels (Methods 1-2). With supervision, we compare weighted voting (WV) and/or filtering with and without PMF. Ranks shown indicate statistically significant differences at $p <= 0.05$ using a two-tailed paired t-test.

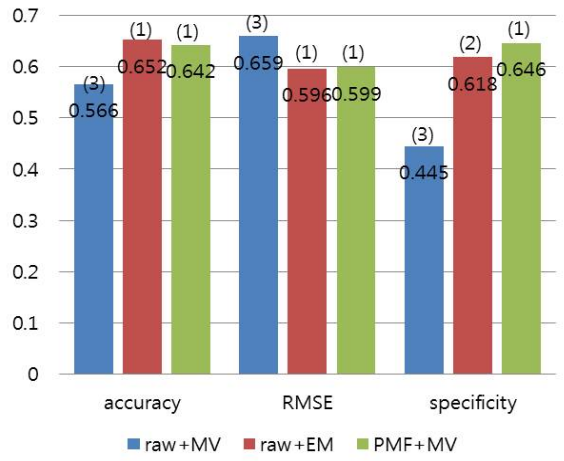

Figure 4: Results of PMF-based inference of missing worker labels over a synthetic data introduced in data set section . Majority voting (MV) with PMF (Method 3) is compared to MV and EM approaches using input (sparse) worker labels (Methods 1-2). Ranks shown indicate statistically significant differences at $p <= 0.05$ using a two-tailed paired t-test.

creases the number of true negatives by reducing the bias of labels based on matrix factorization.

Turk Data Table 1 reports accuracy (ACC), RMSE, and specificity (SPE) achieved by each method over NIST ground truth. In unsupervised setting, PMF with majority voting (Method 3) outperforms the MV baseline (Method 1) in terms of given measures. It even performs equivalently to EM (Method 2). While supervised methods tend to dominate, unsupervised EM and PMF both match performance of the supervised weighted voting (WV) method without filtering.

In supervised setting, worker filtering is clearly seen to provide the greatest benefit, and surprisingly performs better without PMF than with PMF (Methods 6 vs. 7). When filtering is used, use of WV is not seen to further improve performance (Methods 5 vs. 6). We do see PMF-based modeling outperform non-PMF modeling when worker filtering is not employed (Methods 7 vs. 4).

Results of PMF-based inference over Waterloo ground truth are slightly different from the previous one over NIST ground truth as shown in table 2. PMF with majority voting (Method 3) outperforms the MV baseline and shows slightly better performance than EM. However, in supervised setting, weighted voting (Method 4) outperforms all the other methods. In addition, PMF with supervision does not improve the quality of consensus labels significantly compared to unsupervised one. In other words, weighted voting and worker filtering are not helpful to PMF based inference.

PMF with majority voting do not accomplish the highest score compared to supervised methods over two test sets. In the absence of ground truth, PMF is equivalent to or slightly better than EM that is the one of popular methods in unsupervised label aggregation methods. However, there is more room to improve the performance of PMF based inference since we only use majority voting for label aggregation and we replaced all labels with inferred labels.

Conclusion

We presented a label inference method based on probabilistic matrix factorization by transforming a crowdsourcing data into collaborative filtering data. It enables us to predict unlabeled and missing labels from crowd-workers. In addition, we induce consensus labels based on this method and compare its quality with the other well-known label aggregation methods in unsupervised and supervised ways.

While unsupervised consensus labeling accuracy with PMF only matched EM performance, there is a room to improve the performance of our proposed method. Since we use a simple majority voting with PMF, we can expect that the performance would be improved by taking account of other label aggregation methods. In addition, we only use inferred labels for label aggregation in this study. However, it would be interesting for using both inferred and raw labels together to induce more reliable consnesus labels.

Apart from consensus accuracy, complete worker judgments inferred by PMF may have further value, such as predicting the best worker to route a given example to for labeling. Such routing has potential to improve labeling accuracy and reduce the total number of labels required.

Intuitively, an accurate worker's empirical label distribution should resemble the actual class prior. This suggests an alternative, more weakly supervised scenario to consider in

Method	Supervised	Worker Labels	Label Aggregation	ACC	Rank	RMSE	Rank	SPE	Rank
1	No	raw (sparse)	MV	0.457	6	0.424	2	0.263	7
2	No	raw (sparse)	EM	0.468	6	0.426	2	0.328	6
3	No	PMF (complete)	MV	**0.482**	**4**	0.434	2	**0.356**	5
4	Yes	raw (sparse)	WV	**0.518**	**1**	0.433	2	**0.643**	**1**
5	Yes	raw (sparse)	Filtering(α=0.67)	0.504	2	**0.410**	**1**	**0.639**	**1**
6	Yes	raw (sparse)	WV & Filtering(α=0.67)	0.506	2	**0.411**	**1**	0.633	1
7	Yes	PMF (complete)	WV & Filtering(α=0.7)	0.487	4	0.427	2	0.542	4

Table 2: Results of PMF-based inference of missing worker labels over Waterloo ground truth. For the unsupervised case, majority voting (MV) with PMF (Method 3) is compared to MV and EM approaches using input (sparse) worker labels (Methods 1-2). With supervision, we compare weighted voting (WV) and/or filtering with and without PMF. Ranks shown indicate statistically significant differences at $p <= 0.05$ using a two-tailed paired t-test.

which class priors are known while example labels are not. In the unsupervised case, we might instead simply examine the distribution of empirical priors for each worker and detect outliers (Jung and Lease 2011). We plan to investigate these ideas further in combination with those described here.

Acknowledgments

We thank Mark Smucker, Catherine Grady, Wei Tang, and Chandra Prakash Jethani for assistance in collecting relevance judgments and evaluation design. We also thank the anonymous reviewers for their valuable feedback. This work was partially supported by an Amazon Web Services award and a John P. Commons Fellowship for the second author.

References

Alonso, O.; Rose, D.; and Stewart, B. 2008. Crowdsourcing for relevance evaluation. *SIGIR Forum* 42(2):9–15.

Buckley, C.; Lease, M.; and Smucker, M. D. 2010. Overview of the TREC 2010 Relevance Feedback Track (Notebook). In *Proc. of the 19th Text Retrieval Conference*.

Chen, S.; Zhang, J.; Chen, G.; and Zhang, C. 2010. What if the irresponsible teachers are dominating? a method of training on samples and clustering on teachers.

Dawid, A. P., and Skene, A. M. 1979. Maximum likelihood estimation of observer error-rates using the EM algorithm. In *Applied Statistics, Vol. 28, No. 1.*

Dekel, O., and Shamir, O. 2009. Vox populi: Collecting high-quality labels from a crowd. In *COLT*.

Ipeirotis, P.; Provost, F.; and Wang, J. 2010. Quality management on amazon mechanical turk. In *Proceedings of the ACM SIGKDD workshop on human computation*, 64–67. ACM.

Jung, H. J., and Lease, M. 2011. Improving Consensus Accuracy via Z-score and Weighted Voting. In *Proceedings of the 3rd Human Computation Workshop (HCOMP) at AAAI*.

Raykar, V.; Yu, S.; Zhao, L.; Valadez, G.; Florin, C.; Bogoni, L.; and Moy, L. 2010. Learning from crowds. *The Journal of Machine Learning Research* 99:1297–1322.

Salakhutdinov, R., and Mnih, A. 2008. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems (NIPS)*, volume 20.

Sheng, V. S.; Provost, F.; and Ipeirotis, P. G. 2008. Get another label? improving data quality and data mining using multiple, noisy labelers. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*.

Smucker, M. 2012. Crowdsourcing with a Crowd of One and Other TREC 2011 Crowdsourcing and Web Track Experiments. In *Proceedings of TREC*.

Snow, R.; O'Connor, B.; Jurafsky, D.; and Ng, A. Y. 2008. Cheap and fast—but is it good? In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Welinder, P., and Perona, P. 2010. Online crowdsourcing: Rating annotators and obtaining cost-effective labels. In *Computer Vision and Pattern Recognition Workshops (CVPRW)*, 25–32.

Whitehill, J.; Ruvolo, P.; Wu, T.; Bergsma, J.; and Movellan, J. 2009. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *Advances in Neural Information Processing Systems (NIPS)*.