Probabilistic Modeling for Crowdsourcing Partially-Subjective Ratings

An T. Nguyen[1] **Matthew Halpern[2]** **Byron C. Wallace[3]** **Matthew Lease[4]**

[1]Dept. of Computer Science, [2]Dept. of Electrical and Computer Engineering, and [4]School of Information
University of Texas at Austin
[1]atn@cs.utexas.edu, {[2]matthalp, [4]ml}@utexas.edu
[3] College of Computer and Information Science, Northeastern University, byron@ccs.neu.edu

Abstract

While many methods have been proposed to ensure data quality for objective tasks (in which a single correct response is presumed to exist for each item), estimating data quality with subjective tasks remains largely unexplored. Consider the popular task of collecting instance ratings from human judges: while agreement tends be high for instances having extremely good or bad properties, instances with more middling properties naturally elicit a wider variance in opinion. In addition, because such subjectivity permits a valid diversity of responses, it can be difficult to detect if a judge does not undertake the task in good faith. To address this, we propose a probabilistic, *heteroskedastic* model in which the means and variances of worker responses are modeled as functions of instance attributes. We derive efficient Expectation Maximization (EM) learning and variational inference algorithms for parameter estimation. We apply our model to a large dataset of 24,132 Mechanical Turk ratings of user experience in viewing videos on smartphones with varying hardware capabilities. Results show that our method is effective at both predicting user ratings and in detecting unreliable respondents.

1 Introduction

Though many methods have been proposed for evaluating data quality and worker performance in crowdsourcing, prior research has typically assumed *purely-objective* tasks in which each question has a single correct answer. At the other extreme, *purely-subjective* (i.e., opinion) tasks permit a wide range of valid responses with little expectation of agreement between individuals (e.g., asking one's favorite color or food). Between these simple extremes, however, lies a wide, interesting, and important space of *partially-subjective* tasks in which answers are only partially-constrained (Tian and Zhu 2012). Such tasks have received scant attention in prior crowdsourcing research, both in characterizing the nature of such subjectivity, as well as in proposing appropriate modeling techniques for evaluating data quality and worker performance in the absence of a simple gold standard for measuring correctness.

In this paper, we consider the popular task of collecting human ratings of instances having varying properties. While people often agree in rating intances with extremely good or bad properties, instances with more middling properties naturally exhibit wider variance in opinion. Also, it can be difficult to detect if a respondent does not undertake a task in good faith when a diversity of responses are plausible.

More specifically, we consider a task in which human judges are asked to provide ordinal ratings of instances on a scale (e.g., from 1-5). While one expects ratings to vary among respondents as a function of item properties and personal preferences, understanding *how* these ratings are distributed with respect to instance properties can be quite valuable. For example, someone developing a new product might want to predict expected customer satisfaction as a function of different product options; how many potential customers would be pleased by a particular configuration, and are customer needs sufficiently diverse to merit offering multiple versions of the product? How much variance is there in consumer perceptions across the configurations offered?

To address such questions, we propose a probabilistic, *heteroskedastic* model (Bishop 2006; Bishop and Quazaz 1997) in which both mean and variance are modeled as functions of instance properties. Our method formalizes a simple intuition: if someone consistently provides ratings that are 'far' from others', that person is probably less reliable. Critically, this notion of 'far' is modeled as varying in respect to each particular configuration of the instance (i.e., the relative subjectivity of a given instance's properties). For instances that are apparently more subjective (the configuration is not clearly good or clearly bad), the judge's reputation will be penalized less if her response deviates from the mean compared to scenarios in which most judges are in agreement and yet she dissents. This is similar to the objective task analog in which greater disagreement on a given question is taken to indicate greater question difficulty, with respondents penalized less for missing more difficult questions (Whitehill et al. 2009). While Tian and Zhu (2012) also addressed subjective tasks, they model subjectivity only through observed responses, with no notion of underlying properties shared across questions. They also make a strong assumption that all workers must answer all questions.

We adopt data from Halpern, Zhu, and Janapa Reddi (2016)'s study in which each instance to be rated is a short video of user interaction with a mobile

application under a particular hardware configuration[1]. Workers watch the video and rate from 1-5 how satisfied they would be with the hardware configuration given the application performance shown in the video. Does the hardware seem sufficiently capable to support smooth interaction with the mobile application (5), or or does the application seem to struggle to run on legacy hardware, with jerky or sluggish performance (1)? The authors sought to study how user satisfaction changes with varying hardware configuration, such as CPU frequency, and what hardware design techniques could improve user experience.

Contributions. We develop a probabilistic approach to model subjective ratings collected from a crowd. We advocate the importance of treating rating response variance as a function of the attributes of the item being rated, i.e., to model heteroskedastic variance. We believe that the notion of heteroskedastic variance may be less familiar to some communities who collect subjective responses from human subjects, and who might otherwise tend to apply standard linear regression to such data today. We show that such simpler variance modeling suffers significant loss of fidelity.

More significantly, we present a model specification and an Expectation Maximixation (EM) learning algorithm. We also present a Bayesian version with efficient parameter estimation via variational inference. To show the merit of our approach, we report extensive experiments on Halpern, Zhu, and Janapa Reddi (2016)'s large dataset of 24,132 Mechanical Turk ratings of smartphone user satisfaction for varying hardware configurations. To facilitate reproducibility and future work, our source code is available online[2].

2 Related work

Whereas subjective opinion (or polling) tasks permit any answer that reflects an honest opinion, objective annotation tasks rely on shared guidelines which steer annotators to produce similar annotations for a given input example. However, every annotation task exhibits some degree of disagreement, even among trusted annotators, with harder annotation tasks intuitively yielding greater disagreement. Hence it is critical with new annotation tasks that multiple-annotation is performed on a subset of the data so that inter-coder agreement can be measured (Artstein and Poesio 2008). This allows one to gauge task difficulty and establish an upper-bound against which reasonable performance expectations can be set for evaluating the performance of noisy algorithms and non-expert crowds (Gurari et al. 2015).

While some work has investigated use of multiple annotations per example in training or evaluation (Sheng, Provost, and Ipeirotis 2008), use of a single label per example is far more common (by either performing only single labeling or resolving disagreement between labels, manually or by automatic aggregation). The extreme assumption of fully-objective tasks is that any disagreement with this gold standard reflects error. Much research on label aggregation operates in this space, modeling worker reliability or accuracy in relation to this gold standard (Dawid and Skene 1979; Sheshadri and Lease 2013). Some work has captured the intuition that annotators will tend to disagree on more difficult examples (Whitehill et al. 2009). Similarly, some work has recognized that individuals bring different biases in how they understand and internalize annotation guidelines, and that we should penalize annotators less for making systematic errors reflecting bias (correctable through calibration) vs. random errors (Ipeirotis, Provost, and Wang 2010).

Some researchers have found it useful to cluster annotators while still subscribing to the notion of a single gold standard (Kajino, Tsuboi, and Kashima 2013; Venanzi et al. 2014; Kovashka and Grauman 2015; Nguyen, Wallace, and Lease 2015). These clusters group annotators with similar abilities or biases to improve estimation and calibration. For example, Kajino, Tsuboi, and Kashima (2013) consider synthetic data with two clusters ("good" vs. "bad" annotators) and real-data having unknown latent clusters. Kovashka and Grauman (2015) pursue the idea of *Schools of Thought* (SoT). However, they still assume that "there is some single, common interpretation of the property shared consistently by all human viewers – namely, that a single ordering of images from least to most [attribute] is possible."

Other work has gone farther to suggest that the notion of a single, universal gold standard becomes problematic when different communities might be expected to exhibit valid disagreement (Sen et al. 2015). For example, one could imagine conservatives vs. liberals disagreeing on a political labeling task, with neither group being more correct. In such a case, while we might aggregate multiple labels within a group to induce group consensus, naively aggregating such multi-modal data across groups would likely induce a gold-standard acceptable to no one. Instead, on such tasks we might recognize a plurality of multiple, distinct gold standards against which algorithms should be benchmarked. In addition, note that this framing still subscribes to the notion of gold standards, but generalizes it from being universal (singleton) to group-based (plural). A similar SoT framing suggests that annotators naturally cluster into different groups based on their expertise and how they perceive a given task (Welinder et al. 2010).

In this work, we focus on the subjective task of relative rating for which disagreement is particularly notorious and examples abound in practice: e.g., judging movie or product quality (Adomavicius and Tuzhilin 2005), movie maturity (Ipeirotis, Provost, and Wang 2010), the similarity of words (Snow et al. 2008), relative attributes of fashion (Kovashka and Grauman 2015), relevance of search results (Zhang et al. 2014), or the quality of an image (Ghadiyaram and Bovik 2016) or hardware configuration (Halpern, Zhu, and Janapa Reddi 2016). HCI and general survey design in marketing/psychology often check self-consistency of responses rather than inter-worker (peer) agreement Zhang et al. (2014). Individuals may disagree with everyone else provided they are at least self-consistent.

In contrast to objective tasks, measures of inter-coder agreement (Artstein and Poesio 2008) and "gold standards" are less meaningful when applied to relative ratings as far less agreement is expected. While it may still be useful to aggregate ratings (e.g., to compute the mean or me-

[1] github.com/Matthalp/mobile-cpu-user-sat-data.git

[2] github.com/thanhan/subjective-crowd-hcomp16

dian), such aggregation is not viewed as having captured the group's consensus, and it masks valuable information about the range of opinions (Ghadiyaram and Bovik 2016). Instead, it is typically more informative and useful to model the actual distribution of individual ratings. Similar to the earlier discussion of SoT, collaborative filtering and recommendation (Adomavicius and Tuzhilin 2005) is also based on grouping similar users, premised on the idea that similar users will like similar items (i.e., rate those items similarly). As in marketing and polling, data mining is often employed to discover groups of consumers or voters exhibiting similar behavior, or predict how a given group might rate a given item (e.g., mean and variance).

Tian and Zhu (2012) assume questions may have multiple, valid answers, but without any notion of common properties shared across questions. As a result, they can discover degree of subjectivity (i.e., variance) only through the number of distinct answers a question receives. They also assume that every worker has labeled every instance, which limits its applicability to typical crowdsourcing settings in which workers freely choose to do as little or much work as they like. In contrast, we make no such requirement of workers, and each rating question in our task is parameterized by a set of shared properties (i.e., a particular hardware configuration). This allows us to model relationships between questions, and to encode prior beliefs (e.g., that extreme configuration questions will be more objective while middling configuration questions will be more subjective).

Related work in recommender systems has investigated 'shilling attack' detection and resilience, where faked user profiles and ratings are inserted (Gunes et al. 2014) to increase/decease the popularity of some instances or to discrupt the system. We do not model unreliable ratings as adversarial, and to our best knowledge, previous work in that area has not used instance-level features, as in our work.

With respect to probabilistic modeling in crowdsourcing, since the classical work of Dawid and Skene (1979), recent work has extended the idea to adopt a Bayesian approach (Liu and Wang 2012; Kim and Ghahramani 2012), classification models (Raykar et al. 2010), worker dynamics (Simpson et al. 2013), communities of workers (Venanzi et al. 2014), instance-level features (Kamar, Kapoor, and Horvitz 2015), and worker demographic features (Lakkaraju et al. 2015). The common theme in these papers is to model the true label for each instance as a hidden variable. Although some models estimate the variance of true labels, critically, these variance estimates measure the model's uncertainty due to limited data; with unlimited data, these variances would go to zero. In contrast, our model's variance estimates captures inherent subjectivity of the instances, permitting non-zero variance regardless of training data size.

3 Method

3.1 Model

In the following, we use i to index instances and j to index workers. Let L_{ij} be the label given by worker j for instance i. In this case, L_{ij} is a rating on a scale from 1 to 5. We assume that for each label L_{ij}, there is a latent variable Z_{ij} indicating whether the label is reliable or not. If $Z_{ij} = 1$ then L_{ij} is reliable and assumed to be a sample from a Normal distribution with mean and standard deviation predicted by two linear models conditioned on the features of instance i (which we will describe shortly). If $Z_{ij} = 0$ then L_{ij} is assumed to be a sample from a fixed distribution, independent of the instance i. Here, we have modeled ordinal ratings by the continuous Normal distribution. Although the Normal distribution is over real numbers, it does capture the order of the ratings and is also easy to work with. This is a common modeling technique, widely used in recommender system, e.g., in Salakhutdinov and Mnih (2008)'s work.

While a simple 'spammer' model might select between possible labels uniformly at random, we speculate that disingenous raters will tend to avoid extreme ratings (e.g., 1 or 5), which could be far off the mark and more easily detected, and instead prefer ratings close to the middle of the scale (e.g., 2-4). We thus assume that given $Z_{ij} = 0$, L_{ij} has a Normal distribution with mean at the rating scale's midpoint.

Another common simplifying modeling assumption is that 100% of a worker's labels are either spam or ham. In reality, a rater can easily alternate between working diligently vs. becoming bored, tired, or otherwise inattentive. To account for this, we assume a parameter θ_j specifying the proportion of disingenous ratings by worker j. For example, $\theta_j = 0.2$ means that worker j generates unreliable labels 80% of the time. This type of modeling technique is often use in discrete choice model in economics (Greene 2009).

These modeling assumptions can be summarized by:

$$Z_{ij} \sim \text{Ber}(\theta_j) \quad (1)$$

$$L_{ij}|Z_{ij} = 1 \sim \mathcal{N}(\mu_i, \sigma_i^2) \quad (2)$$

$$L_{ij}|Z_{ij} = 0 \sim \mathcal{N}(3, s) \quad (3)$$

where 3 splits the 1-5 rating range, s is the unreliable label variance; μ_i and σ_i are the mean and standard deviation of reliable ratings of instance i, predicted by two linear models:

$$\mu_i = \mathbf{w}^T \mathbf{x}_i \quad (4)$$

$$\sigma_i = \exp(\mathbf{v}^T \mathbf{x}_i) \quad (5)$$

where $\mathbf{x}_i$ is the feature vector for instance i and $\mathbf{w}$ and $\mathbf{v}$ are parameters to learn (the exp function is used to make sure σ_i is positive). By modeling the variance as function of the instance features, we have made the *heteroskedastic* assumption. This is in contrast with the *homoskedastic* assumption in standard linear models that assume a fixed variance.

Figure 1 shows the factor graph for our model. This depicts the (partially) Bayesian variant, which we present in further detail in Section 3.3.

3.2 Learning

To learn the parameters ($\mathbf{w}, \mathbf{v}, s$ and θ), we use Expectation Maximization (EM) (Dempster, Laird, and Rubin 1977). The E-step infers the posterior distribution over the hidden variables $\{Z_{ij} \forall i, j\}$ by evaluating (the right hand side of):

$$\Pr(Z_{ij} = 1|L) \propto \mathcal{N}(L_{ij}|\mu_i, \sigma_i) \cdot \text{Ber}(Z_{ij} = 1|\theta_j) \quad (6)$$

$$\Pr(Z_{ij} = 0|L) \propto \mathcal{N}(3, s) \cdot \text{Ber}(Z_{ij} = 0|\theta_j) \quad (7)$$

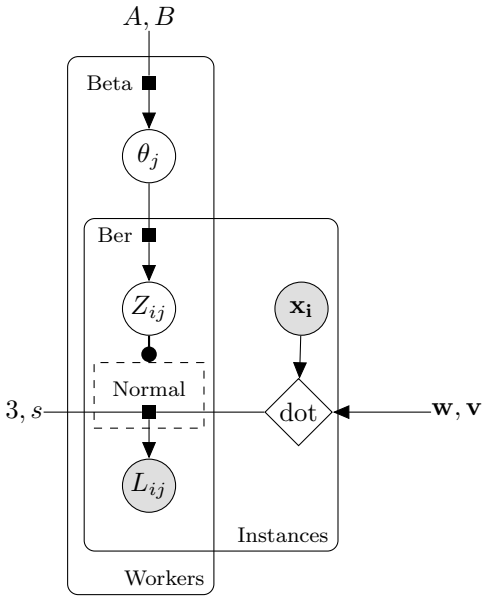


Figure 1: The factor graph for our model. Circles represent random variables (shaded ones are observed), squares represent factors, diamonds are deterministic functions, edge endpoints (μ, C, θ) are unknown constants or parameters, plates denote repetitions, and dotted plates are gates. In this case, the value of Z_{ij} is used to select the parameters for the Normal distribution; see Equations 2 and 3. We elide the transformation (exp) function applied to $\mathbf{v^T x}$ for simplicity.

and normalizing these two quantities to obtain $t_{ij} = \Pr(Z_{ij} = 1|L)$, where t_{ij} is the posterior probability that the label for instance i by worker j is reliable.

Let $\sum_{ij}$ be the sums over all instances and all labels for each instance. In the M-step, we want to maximize the expected complete data log likelihood (where the expectation is with respect to the posterior computed in the E-step):

$$\begin{aligned} Q(\mathbf{w}, \mathbf{v}) &= \sum_{ij} \mathbb{E}_{Z_{ij}} \cdot l_{ij}(\mathbf{w}, \mathbf{v}, z_{ij}) \\ &= \sum_{ij} t_{ij} \cdot l_{ij}(\mathbf{w}, \mathbf{v}, 1) + (1 - t_{ij}) \cdot l_{ij}(\mathbf{w}, \mathbf{v}, 0) \end{aligned} \tag{8}$$

where l_{ij} is the contribution to the likelihood of a label:

$$\begin{aligned} &l_{ij}(\mathbf{w}, \mathbf{v}, z_{ij}) = \log \mathrm{Ber}(Z_{ij} = z_{ij}|\theta_j) + \\ &\mathbb{I}(z_{ij} = 1) \cdot \log \mathcal{N}(L_{ij}, \mu_i, \sigma_i^2) + \\ &\mathbb{I}(z_{ij} = 0) \cdot \log \mathcal{N}(3, s) \end{aligned} \tag{9}$$

The gradients of Q with respect to $\mathbf{w}$ and $\mathbf{v}$ can be evaluated as follows:

$$\frac{\partial Q}{\partial \mathbf{w}_k} = \sum_{i,j} \frac{L_{ij} - \mathbf{w}^T x_i}{\sigma_{ij}^2} x_{ik} t_{ij} \tag{10}$$

$$\frac{\partial Q}{\partial \mathbf{v}_k} = \sum_{ij} \left(-x_k + \frac{(L_{ij} - \mathbf{w}^T x_i)^2}{\sigma_{ij}^2} x_{ik} \right) t_{ij} \tag{11}$$

For the actual maximization of $\mathbf{w}$ and $\mathbf{v}$, we us the BFGS algorithm (experiments with stochastic gradient descent were faster but resulted in worse performance). To estimate the parameter s, we simply evaluate the weighted sample variance of all the labels, where the weight for a label is the posterior that the label is unreliable:

$$s = \frac{\sum_{i,j}(1 - t_{ij})(L_{ij} - 3)^2}{\sum_{ij}(1 - t_{ij})} \tag{12}$$

3.3 Bayesian Model of Worker Reliability

Our method so far has assumed a fixed parameter θ_j for each worker j. In practice, we would like to have a measure of 'confidence' for each estimate of θ. For example, suppose worker 1 has labeled 10 instances and worker 2 has labeled 100 instances. In this case we should be more confident regarding θ_2 than θ_1. The Bayesian framework provides a principled way to quantify the confidence in our estimates. Furthermore, a distributional emtimate would make the model more robust to overfitting. We thus adopt a Bayesian treatment of θ_j, introducing a prior:

$$\theta_j \sim \mathrm{Beta}(A, B) \tag{13}$$

After assuming this prior distribution over θ_j, we observe an interesting analogy between our model and Latent Dirichlet Allocation (LDA) (Blei, Ng, and Jordan 2003), which models 'topics' in discrete data such as text. Roughly speaking, a worker corresponds to a document, a label corresponds to a word, and a *School of Thought* (SoT) (Tian and Zhu 2012) corresponds to a topic. Our model here assumes two SoT: reliable and unreliable, but the model can be extended to include additional 'schools'. Our model's generation of workers and allocation of SoTs to workers are the same as LDA's generation of documents and allocation of topics. Our label generation, though, differs from LDA's word generation.

Unfortunately, with the priors on θs, closed-form EM updates are not possible. We develop a meanfield variational approach similar to Blei, Ng, and Jordan (2003)'s. Variational inference (Wainwright and Jordan 2008) aims to approximate the posterior distribution over all hidden variable given the data, which in our model is $p(\mathbf{z}, \boldsymbol{\theta}) = \Pr(\mathbf{z}, \boldsymbol{\theta}|L)$, where $\mathbf{z}, \boldsymbol{\theta}$ and $\mathbf{L}$ are vectors of Z_{ij}, θ_j and L_{ij} $\forall i, j$. The idea is to introduce a function over the same variables: $q(\mathbf{z}, \boldsymbol{\theta})$ and perform optimization to minimize the KL divergence $\mathbf{KL}(q||p)$, making q 'close' to p. By minimizing this divergence, we maximize a lower bound on the data log likelihood. For the optimization to be efficient, q should have a simple form. The meanfield assumption is that q factorizes:

$$q(\mathbf{z}, \boldsymbol{\theta}) = \prod_{ij} q(Z_{ij}) \prod_{j} q(\theta_j) \tag{14}$$

We further assume the forms of the factor functions q (disambiguated by argument):

$$q(Z_{ij}) = \mathrm{Ber}(\gamma_{ij}) \tag{15}$$

$$q(\theta_j) = \mathrm{Beta}(\alpha_j, \beta_j) \tag{16}$$

where γ_{ij}, α_j and β_j are the variational parameters that should be selected to minimize the above KL divergence.

This optimization problem can be solved via coordinate descent: optimizing (or updating) one factor while keeping all others fixed and iterate until convergence. By following the meanfield formulation (see, for example, (Murphy 2012, sec 21.3)), we can derive the corresponding update equations. For Z_{ij}, we have:

$$q^*(Z_{ij}=0) \propto \mathcal{N}(3,s) \cdot \exp\{\mathbb{E}_{q(\theta_j)} \log(1-\theta_j)\} \quad (17)$$

$$q^*(Z_{ij}=1) \propto \mathcal{N}(L_{ij}|\mu_i,\sigma_i) \cdot \exp\{\mathbb{E}_{q(\theta_j)} \log(\theta_j)\} \quad (18)$$

The expectations in these equations can be evaluated efficiently as follows:

$$\mathbb{E}_{q(\theta_j)} \log(\theta_j) = \Psi(\alpha_j) - \Psi(\alpha_j+\beta_j) \quad (19)$$

$$\mathbb{E}_{q(\theta_j)} \log(1-\theta_j) = \Psi(\beta_j) - \Psi(\alpha_j+\beta_j) \quad (20)$$

where Ψ is the Digamma function. To update the factors on θ, we have:

$$\begin{aligned} q^*(\theta_j) &\propto \mathrm{Beta}(A_j, B_j) \\ &\quad \exp\left\{\sum_i \mathbb{E}_{q(z_{ij})} \log \mathrm{Ber}(Z_{ij}|\theta_j)\right\} \\ &\propto \mathrm{Beta}(A_j + \sum_i q(Z_{ij}=0), \\ &\quad B_j + \sum_i q(Z_{ij}=1)) \end{aligned} \quad (21)$$

Equation 17 and 18 together define the update for each hidden variable Z_{ij}, which identifies whether the label for instance i by worker j is reliable. This update is based on the likelihood of that label and the current approximation of worker j quality (which is θ_j). Equation 21 updates the approximation over θ_j by starting from the Beta prior, considering all the labels provided by worker j, and then summing up the current approximations over the indicators Z_{ij}. In our implementation, we repeatedly apply these update equations until the average change in the parameters is less than 0.01.

The variational inference procedure described above plays the role of the E-step in a variational EM approach. The M-step can be performed in a similar manner as in the previous section.

4 Evaluation

4.1 Evaluation Setup

Dataset We use a large dataset of Mechanical Turk ratings collected by Halpern, Zhu, and Janapa Reddi (2016). This dataset consists of 370 hardware *configurations*. Each configuration (modeled as an instance) is a particular mobile application (one of 13) running on a given CPU and GPU. For example, one included configuration runs the 'YouTube' application on a CPU with a frequency 2.5 GHz, with 3 cores enabled, and with a GPU frequency of 320 MHz. For each of 7 "Current Generation" applications, there are 28 configurations, while each of 6 "Next Generation" applications have 29 configurations (7x28 + 6x29 = 370 configurations).

For each configuration, a short video (about one minute) of an user interaction is recorded. For instance, the user opens YouTube, does a search and plays a clip. Workers are asked to watch the video and provide a rating from 1 to 5, indicating how satisfied they would be with the hardware configuration given the performance of the application shown in the video. Does the hardware seem sufficiently capable to support smooth interaction with the mobile application (5: "very satisfied"), or or does the application seem to struggle to run on legacy hardware, with jerky or sluggish performance (1: "very unsatisfied")? Approximately 60 different worker ratings were collected for each configuration, totaling 24,132 ratings collected from 3,241 workers.

We expect that for better configurations (high frequencies, more cores), the user will be more satisfied and the average ratings will be higher. Furthermore, for configurations that are extremely good or bad, the workers will tend to agree with one another more often. For the configurations having middling properties, we expect greater disagreement, or higher variance (of the ratings distribution).

Configurations are represented by the feature vector:

$$\mathbf{x}_i = [\mathbb{I}(\text{Angry Bird}), ..., \mathbb{I}(\text{Youtube}), \text{CPU freq}, \text{ Cores}, \text{ GPU freq}]$$

By including 13 indicators (one for each application), we can make effective predictions across applications. Additional features used in prediction are the CPU frequency, the number of cores enabled, and the GPU frequency. We further experimented with more complex features, including polynomial and interaction terms, but these experiments showed very similar results, so we omit reporting them.

We randomly split the 370 configurations into 60% training data, 20% valildation data, and 20% test data. We do not control for applications in our sampling, thus we expect all 13 applications to be represented in each division of the data. In Section 4.3, we experiment with sub-sampling the training data further to investigate prediction accuracy under sparser training conditions.

Methods We evaluate the two different versions of our heteroskedastic model:

1. **NEW**: the original model with EM learning (Section 3.2).
2. **B-NEW**: the Bayesian model with variational inference (Section 3.3).

We initalize parameters W and V using parameters of LR2 (see Section 4.3). We use the validation set to tune the three hyper-parameters by searching in appropriate ranges (see **Table 1**). The first two (n_V and n_W) are a form of early stopping that prevents overfitting of the parameters (W and V). The last one (h_θ) is the inital value for the parameter θ_j for each worker j, which represents an initial guess of worker j's reliability. For the Bayesian model, we reuse the same ($n_V = 1$ and $n_W = 3$) and set the Beta prior to $A = 8$ and $B = 2$ based on the final value $h_\theta = 0.8$ (recall that $\mathrm{Beta}(A, B)$ is the prior belief of each worker's reliability).

4.2 Task 1: Prediction with Reliable Raters

Our first set of experiments assume that our human subjective raters are entirely reliable. In this case, we do not need to

Hyper-parameters	Search Range	Final value
n_V: number of EM iterations that update V	$\{1, 2, 3\}$	1
n_W: number of EM iterations that update W	$\{1, 2, 3\}$	3
h_θ: initial value for θ_j, for each worker j	$\{.7, .8, .9, .95\}$	0.8

Table 1: Hyper-parameters are tuned on validation data based on search ranges. Final values are shown.

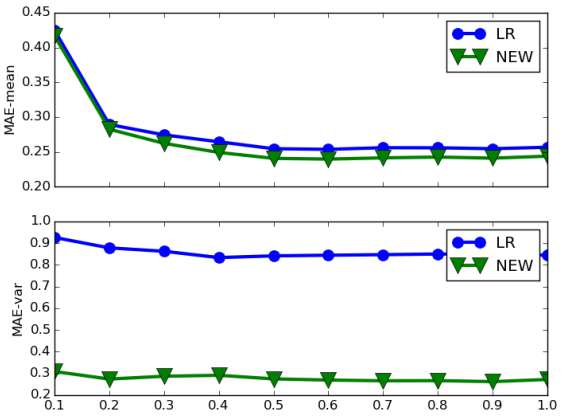


Figure 2: Mean Absolute Error (MAE, *lower is better*) on the test set with respect to mean (top) and variance (bottom). Along the x-axis, we vary the proportion of configurations used for training, e.g., with half of the configurations used for model training at $x = 0.5$. LR is (homoskedastic) Linear Regression, whereas NEW assumes heteroskedastic variance. Results are shown after averaging over 5 runs.

detect any unreliable workers, but instead can focus our effort entirely on predicting the means and variances of unseen configurations in our partially-subjective modeling task.

Given a training set of configurations with labels, for each configuration in an unlabeled test set, we wish to predict the mean and variance of its ratings. As motivated in Section 1, while one expects ratings to vary among respondents as a function of item properties and personal preferences, one needs to know specifically *how* these ratings are distributed with respect to instance properties. For example, someone developing a new product might like to gauge expected customer satisfaction as a function of different product options, understanding how many potential customers would be pleased by a particular configuration, or whether customers and their needs are sufficiently diverse to merit shipping multiple versions of the product. How much variance is there in consumer perceptions?

Evaluation Metric. We adopt Mean Absolute Error (MAE) of the predicted means and variances compared to the gold values (in the test set). These gold values are computed as the empirical means and variances of the ratings observed in the dataset for each configuration.

Baseline we expect that simple Linear Regression **(LR)** is very likely to be the most common method that many researchers and practioners would turn to today, due to its simplicity, familiarity (from being covered in most statistics courses), and continuing widespread use in both practice (industrial and clinical) and in published research studies.

However, standard LR implicitly makes a strong assumption of *homoskedasticity* (i.e., that variance in user ratings remains constant across configurations). This assumption is extremely restrictive with subjective tasks, such as our partially-subjective scenario, because instances will differ in their subjectivity as a function of their configurations. To show the importance of this issue in modeling the variance of our data, we compare the heteroskedastic approach vs. LR's homoskedasticity for this estimation task.

Because this first experiment assumes that human raters are entirely reliable, we utilize a simpler version of our heteroskedastic approach which does not model unreliable workers and labels. Specifically, we set all indicators Z_{ij} to 1 (implying that all labels are reliable and effectively 'disabling' our reliability model). We show that heteroskedastic modeling provides far better variance estimates while modestly improving the mean estimates as well.

Figure 2 presents results of comparing homoskedastic LR to the simplified version of our heteroskedastic model. In the top sub-plot of **Figure 2**, one can observe that the latter assumption improves prediction of the means. Although the mean prediciton model in our method is essentially LR, accounting for the variance in the variances has improved its performance. However, results with respect to variance prediction (bottom sub-plot) clearly indicate that the constant variance estimated by homoskedastic LR is a very poor fit for the subjective ratings in our data. In contrast, the heteroskedastic nature of our our method enables it to effectively model the different variances for different configurations. With such a striking difference, the conclusion seems clear: it is critical to account for heteroscedasticity in worker responses when modeling such subjective tasks.

4.3 Task 2: Prediction with Unreliable Raters

As motivated in Section 1, *partially-subjective tasks* represent a wide, interesting, and important class of tasks in which answers are partially-constrained (Tian and Zhu 2012). While we can expect most workers will undertake tasks in good faith, some may not. Detecting such cases becomes vastly more difficult with subjective tasks because the questions posed permit a valid diversity of responses.

Given a set of labeled configurations with some unknown proportion of unreliable workers, we wish to predict the means and variances, as in *Task 1: Prediction with Reliable Raters* (Section 4.2). However, we also wish to detect unreliable workers. The output for this detection task is, for each worker, the probability that the worker is unreliable.

However, we face a fundamental challenge in pursuing

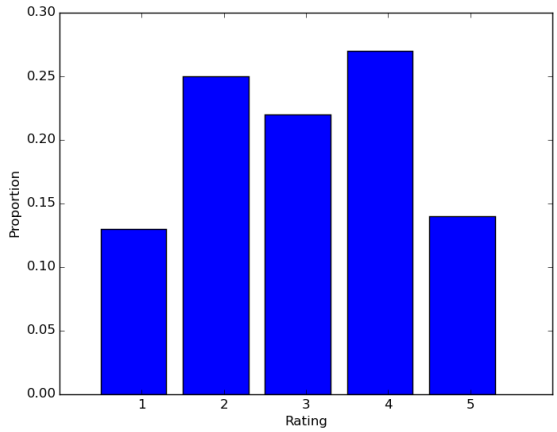


Figure 3: The *Unreliable Label Distribution* is collected empirically by asking workers to "pretend to be unreliable and rate a video without watching it." Interestingly, those pretending to do the task prefer 2 and 4 ratings to 3.

this new line of research: when crowdsourcing "in the wild", we do not know who the unreliable workers are. We therefore simulate unreliable workers as realistically as possible and evaluate under a wide range of possible conditions that could be of practical interest.

Unreliable Label Distribution To simulate unreliable workers as faithfully as possible, we posted a Mechanical Turk task in which workers were asked to *pretend to be unreliable* and enter a false rating between 1 and 5, as if they were performing the task but not really doing the work. 10 workers provided 20 labels each, producing 200 labels in total. **Figure 3** shows the resulting empirial distribution.

As expected, workers generally try to avoid the extreme ratings of 1 and 5. More surprising, however, is that the proportion of 3s is slightly lower than 2s and 4s. Thus, as in practice, our model must contend with actual observed behavior differeing from its modeling assumptions (of normality), with unreliable labels generated from the crowd's observed empirical distribution of false ratings.

Simulation To generate a 'gold standard' of unreliable workers and to explore our method's performance under varying conditions, we simulate unreliable workers and their labels in the training set. Concretely, we consider five evaluation parameters that specify the simulation and evaluation:

1. *Proportion of Unreliable Workers* (default 0.1). For the default value, 10% of the workers are randomly selected to be replaced with 'unreliable' workers, with ratings generated according to the following item below.
2. *Proportion of Unreliable Ratings* (default 0.8). For the default value, an unreliable worker provides an unreliable rating 80% of the time. In our simulation, for each worker that is deemed unreliable, 80% of her ratings are randomly selected and replaced by samples from the 'Unreliable Label Distribution' (described below).
3. *Proportion of Configurations Trained On* (default 1.0). The proportion of the configurations in the train set that are given to the method for training. Recall that the training set consists of 60% of the configurations; while the validation and test sets, 20% each, are held out. By default, all training set configurations are observed.
4. *Proportion of Ratings per Configuration* (default 1.0). For each configuration, this is the proportion of the ratings that are provided during training. The dataset has roughly 60 ratings per configuration; thus 0.2 implies that roughly 12 ratings are randomly selected for each configuration.
5. *Proportion of Workers Removed* (default 0.0). Because this is a real dataset, there may be a number of unknown unreliable workers and labels. This parameter aims to allow us to investigate to what degree these unknown workers and labels affect our results. The goal is to remove workers likely to be unreliable, i.e., those with high *Average Deviation*, which we define below. This parameter specifies the proportion removed. Setting it to 0.05 removes the top 5% of the most unreliable workers, as ranked by Average Deviation, from all of the training, validation and test sets.

Table 2 summarizes evaluation parameters for this task. Parameters 1-2 explore our method's performance under varying unreliable workers and labels. Parameters 3-4 consider sparser data conditions, where a smaller number of configurations (or labels per configuration) is available for training. Parameter 5 tries to 'clean' the data in a preprocessing step to investigate the effect of unknown unreliable workers and labels. For predicting the means and variances of configurations in the test set, we compare the predictions with the empirical means and variances of the labels. For detecting unreliable workers, we directly evaluate the AUC of detection with respect to the simulated unreliable workers.

Baseline For predicting mean and variance, having shown the limitations of homoskedastic modeling of variance in our first experiment (Section 4.2), we now adopt a stronger, heteroskedastic baseline for these experiments. Our baseline comprises two independent Linear Regression models (our experiments using Support Vector Regression produced similar results). Concretely, for each configuration, we compute the empirical mean and variance of its rating labels. Two Linear Regression models are then trained, one for predicting the mean and one for predicting the variance from the configuration features. We refer to this baseline as **LR2**.

For detecting unreliable workers, Area Under the Curve (AUC) of the ROC curve is used as the evaluation metric. Average Deviation (AD) is used as the baseline, as we discuss below. This is a simple and strong baseline. The limitations it does have are as follows: (1) it penalizes workers regardless to the subjectivity of the instance; (2) it approximates the true ratings mean by the average; and, (3) it does not consider the number of ratings a worker has provided.

#	Parameter	Default	Experimental Range
1	Proportion of Unreliable Workers	10%	0%, 5%, 10%, 15%, 20%
2	Proportion of Unreliable Ratings	80%	20%, 40%, 60%, 80%, 100%
3	Proportion of Configurations Trained On	100%	20%, 40%, 60%, 80%, 100%
4	Proportion of Ratings per Configuration	100%	20%, 40%, 60%, 80%, 100%
5	Proportion of Workers Removed	0%	0%, 5%, 10%, 15%, 20%

Table 2: Experimental parameters and their values varied for *Task 2: Prediction with Unreliable Raters*.

Average Deviation (AD). A simple way to measure the quality of a label is to look at its deviation (or absolute difference) from the average label (of the configuration). To measure the quality of a worker, we can simply take the average of the deviations of his labels, expecting quality workers to show low AD. We use this measure in the preprocessing step and also as a baseline for unreliable worker detection.

Results Summary We report results of five experiments. In each, we vary one experimental parameter from **Table 2** while keeping others at their default values. We repeat each experiment five times and report the average.

The EM-Based consistenly performs far better than the LR2 baseline across all three cases of estimating means, estimating variances, and detecting unreliable workers. The benefit of joint modeling is clearly evidenced.

Comparing the Bayesian model (B-NEW) to NEW is more mixed. B-NEW's strength lies in achieving lower error in predicting the variances, and it achieves comparable error in predicting means. However, its performance is worse in detecting unreliable workers (and sometimes even worse than the LR2 baseline). By representing workers by distributions (instead of point estimates), B-NEW is more robust to overfitting of the parameters W and V, hence improving variance prediction. However, B-NEW's priors on workers introduce a bias vs. unreliable workers, and thereby detection performance suffers. Given the different relative strengths of NEW and B-NEW, best performance could be achieved by using the two methods in tandem.

Detailed Results **Figure 4a** shows results of varying experimental Parameter 1: *Proportion of Unreliable Workers*. We find that with higher proportions of unreliable workers, the Linear Regression 2 (LR2) baseline deteriorates quickly while our model maintains good performance (except that B-NEW is slightly worse in the unreliable worker detection task; we discuss this further below). For the case when no unreliable workers are simulated, our method is still as good as the baseline for variance and slightly better for mean. The baseline predicts mean and variance seperately. By using a joint model, our method has improved the mean prediction.

In **Figure 4b**, we present results in the same format observed when varying experimental Parameter 2: the *Proportion of Unreliable Ratings*: how often an unreliable worker provides an unreliable label. We observe a similar pattern for the task of predicting mean and variance in the top and middle plot. For the second task, the results are intuitive: workers that give unreliable labels more often are easier to detect using any method.

In the next two experiments, we varied experimental Parameter 3: *Proportion of Configurations Trained On* (the proportion of configurations available for training the model (**Figure 4c**), and experimental Parameter 4: *Proportion of Ratings per Configuration*, the proportion of labels available per configuration (**Figure 4d**). Overall, while methods perform better with more data, as expected, our model also tends to achieve higher improvement over the baselines.

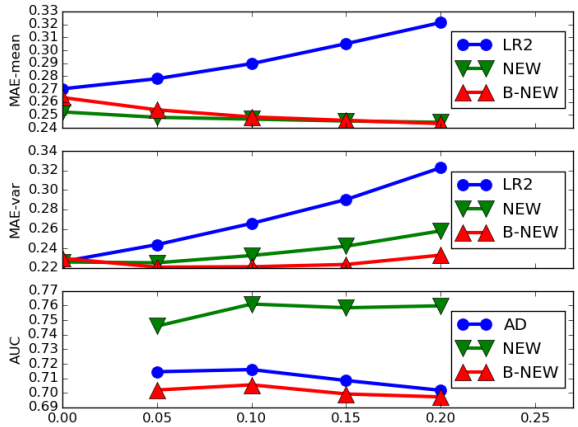
In the last experiment, we vary experimental Parameter 5: *Proportion of Workers Removed*, removing workers with high AD in a preprocessing step (for train, validation and test sets). From the plots in **Figure 4e**, we see that although the results change slightly with varying the number of workers removed, the order of the methods remains the same. This suggests that the unknown unreliable workers in the dataset do not (greatly) affect the results reported earlier.

5 Conclusion

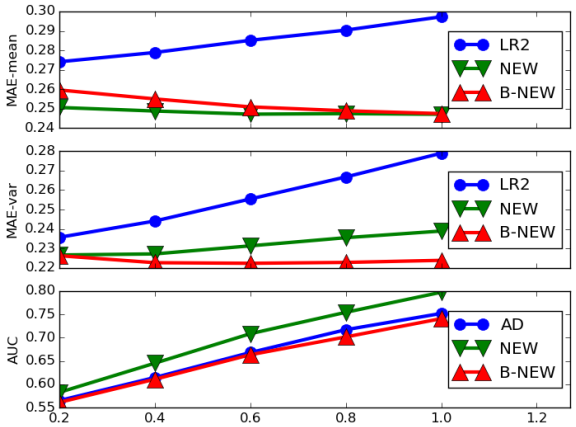
We have identified a new problem of estimation and quality assurance for crowdsourcing *partially-subjective* tasks. We presented a novel method which makes a heteroskedastic assumption, defining a rating generation model that distinguishes between reliable vs. unreliable workers and their ratings. We derived an efficient EM algorithm and a variational inference procedure for the model. In empirical evaluations we found the method to consistenly performs far better than a strong baseline across all three cases of estimating means, estimating variances, and detecting unreliable workers. The benefit of joint modeling is clearly evidenced. Moreover, we showed that the Bayesian variant of our model can be used to further improve the prediction of variance.

Although our evaluation is limited to one dataset, our method can be easily generalized to other rating datasets. A fundamental challenge in pursuing this new research problem for partially-subjective tasks was it involved crowdsourcing "in the wild", where we do not know who the unreliable workers are. While we do our best to realistically simulate unreliable workers and evaluate under a wide range of possible conditions of interest, we would like to pursue further data collection to better identify and characterize unreliable working behaviors for further realism.

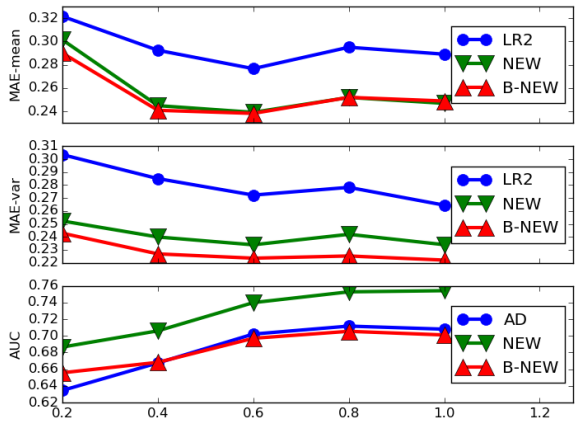
In additional future work, we would like to investigate better learning algorithms, such as an empirical Bayes approach that learns a good prior. We are also interested in using our model to visualize Schools of Thought (Tian and Zhu 2012) in the data, similar to LDA topic model (Blei, Ng, and Jordan 2003). Extension to product ratings data with adversarial ratings would also be interesting to pursue.



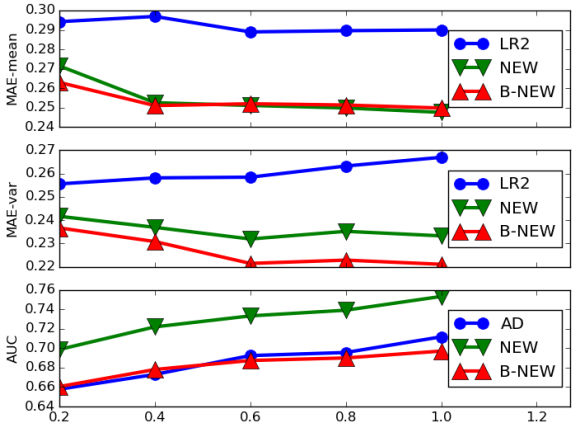
(a) Varying the Proportion of Unreliable Workers.



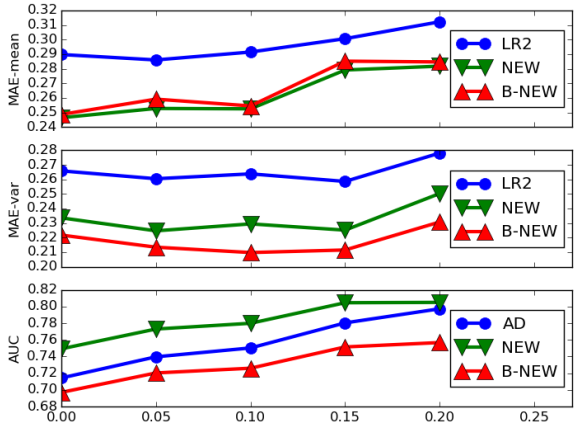
(b) Varying the Proportion of Unreliable Ratings.



(c) Varying the Proportion of Configurations Trained On.



(d) Varying the Proportion of Ratings per Configuration.



(e) Varying the Proportion of Workers Removed.

Figure 4: Results of varying the five different experimental parameters from **Table 2** (on the x-axis). Each figure includes 3 plots. **Top**: Mean Absolute Error (MAE) for mean (*lower is better*). **Middle**: MAE for variance. **Bottom**: Area Under Curve (AUC) for unreliable worker detection (*higher is better*). All results are reported after averaging over 5 runs.

Acknowledgments. We thank the anonymous reviewers for their valuable feedback, and the crowd for their participation, without which our study would not be possible. This study was supported in part by National Science Foundation grant No. 1253413 and IMLS grant RE-04-13-0042-13. Any opinions, findings, and conclusions or recommendations expressed by the authors are entirely their own and do not represent those of the sponsoring agencies.

References

Adomavicius, G., and Tuzhilin, A. 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *Knowledge and Data Engineering, IEEE Transactions on* 17(6):734–749.

Artstein, R., and Poesio, M. 2008. Inter-coder agreement for computational linguistics. *Computational Linguistics*.

Bishop, C. M., and Quazaz, C. S. 1997. Regression with input-dependent noise: A bayesian treatment. In *Advances in Neural Information Processing Systems*, 347–353.

Bishop, C. M. 2006. *Pattern recognition and machine learning*.

Blei, D. M.; Ng, A. Y.; and Jordan, M. I. 2003. Latent dirichlet allocation. *the Journal of machine Learning research*.

Dawid, A. P., and Skene, A. M. 1979. Maximum likelihood estimation of observer error-rates using the em algorithm. *Applied statistics* 20–28.

Dempster, A. P.; Laird, N. M.; and Rubin, D. B. 1977. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society*.

Ghadiyaram, D., and Bovik, A. C. 2016. Massive online crowdsourced study of subjective and objective picture quality. *Image Processing, IEEE Trans. on* 25(1):372–387.

Greene, W. 2009. Discrete choice modeling. In *Palgrave handbook of econometrics*. Springer.

Gunes, I.; Kaleli, C.; Bilge, A.; and Polat, H. 2014. Shilling attacks against recommender systems: a comprehensive survey. *Artificial Intelligence Review* 42(4):767–799.

Gurari, D.; Theriault, D.; Sameki, M.; Isenberg, B.; Pham, T. A.; Purwada, A.; Solski, P.; Walker, M.; Zhang, C.; Wong, J. Y.; et al. 2015. How to collect segmentations for biomedical images? a benchmark evaluating the performance of experts, crowdsourced non-experts, and algorithms. In *Applications of Computer Vision (WACV), IEEE Winter Conference on*.

Halpern, M.; Zhu, Y.; and Janapa Reddi, V. 2016. Mobile cpus rise to power: Quantifying the impact of generational mobile cpu design trends on performance, energy, and user satisfaction. In *High Performance Computer Architecture (HPCA), 2015 IEEE 22nd International Symposium on*.

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

Kajino, H.; Tsuboi, Y.; and Kashima, H. 2013. Clustering crowds. In *AAAI Conference on Artificial Intelligence*.

Kamar, E.; Kapoor, A.; and Horvitz, E. 2015. Identifying and accounting for task-dependent bias in crowdsourcing. In *3rd AAAI HCOMP Conference*.

Kim, H.-C., and Ghahramani, Z. 2012. Bayesian classifier combination. In *AISTAT*.

Kovashka, A., and Grauman, K. 2015. Discovering attribute shades of meaning with the crowd. *International Journal of Computer Vision* 114(1):56–73.

Lakkaraju, H.; Leskovec, J.; Kleinberg, J.; and Mullainathan, S. 2015. A bayesian framework for modeling human evaluations. In *SIAM International Conference on Data Mining*.

Liu, C., and Wang, Y.-m. 2012. Truelabel+ confusions: A spectrum of probabilistic models in analyzing multiple ratings. In *ICML*.

Murphy, K. P. 2012. *Machine learning: a probabilistic perspective*. MIT press.

Nguyen, A. T.; Wallace, B. C.; and Lease, M. 2015. Combining Crowd and Expert Labels using Decision Theoretic Active Learning. In *Proceedings of the 3rd AAAI Conference on Human Computation (HCOMP)*, 120–129.

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

Salakhutdinov, R., and Mnih, A. 2008. Bayesian probabilistic matrix factorization using markov chain monte carlo. In *ICML*.

Sen, S.; Giesel, M. E.; Gold, R.; Hillmann, B.; Lesicko, M.; Naden, S.; Russell, J.; Wang, Z. K.; and Hecht, B. 2015. Turkers, scholars, arafat and peace: Cultural communities and algorithmic gold standards. In *ACM Conference on Computer Supported Cooperative Work & Social Computing*.

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*.

Sheshadri, A., and Lease, M. 2013. Square: A benchmark for research on computing crowd consensus. In *First AAAI Conference on Human Computation and Crowdsourcing*.

Simpson, E.; Roberts, S.; Psorakis, I.; and Smith, A. 2013. Dynamic bayesian combination of multiple imperfect classifiers. In *Decision Making and Imperfection*. Springer. 1–35.

Snow, R.; O'Connor, B.; Jurafsky, D.; and Ng, A. Y. 2008. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *EMNLP*, 254–263.

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*, 226–234. ACM.

Venanzi, M.; Guiver, J.; Kazai, G.; Kohli, P.; and Shokouhi, M. 2014. Community-based bayesian aggregation models for crowdsourcing. In *Proceedings of the 23rd international conference on World wide web*, 155–164. ACM.

Wainwright, M. J., and Jordan, M. I. 2008. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning* 1(1-2):1–305.

Welinder, P.; Branson, S.; Perona, P.; and Belongie, S. J. 2010. The multidimensional wisdom of crowds. In *NIPS*.

Whitehill, J.; Wu, T.-f.; Bergsma, J.; Movellan, J.; and Ruvolo, P. 2009. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *NIPS*.

Zhang, Y.; Zhang, J.; Lease, M.; and Gwizdka, J. 2014. Multidimensional relevance modeling via psychometrics and crowdsourcing. In *International ACM SIGIR conference on Research and Development in Information Retrieval*.