Reliable aggregation of boolean crowdsourced tasks

Luca de Alfaro and **Vassilis Polychronopoulos** and **Michael Shavlovsky**
{luca, vassilis, mshavlov}@cs.ucsc.edu
Department of Computer Science
University of California, Santa Cruz

Abstract

We propose novel algorithms for the problem of crowdsourcing binary labels. Such binary labeling tasks are very common in crowdsourcing platforms, for instance, to judge the appropriateness of web content or to flag vandalism. We propose two unsupervised algorithms: one simple to implement albeit derived heuristically, and one based on iterated bayesian parameter estimation of user reputation models. We provide mathematical insight into the benefits of the proposed algorithms over existing approaches, and we confirm these insights by showing that both algorithms offer improved performance on many occasions across both synthetic and real-world datasets obtained via Amazon Mechanical Turk.

Introduction

Crowdsourcing is now in wide use by organizations and individuals allowing them to obtain input from human agents on problems for which automated computation is not applicable or prohibitively costly. Due to the involvement of the human factor, several challenges come with the use of crowdsourcing. Poor quality feedback from users is common due to malevolence or due to misunderstanding of tasks.

Crowdsourcing applications address the problem of poor human workers reliability through redundancy, that is, by assigning the same task to multiple workers. Redundancy comes at a cost: crowdsourced tasks usually cost money, and the use of multiple workers entails an indirect cost of latency in completing the tasks. This raises the issue of how to optimally aggregate the worker's input. Simple majority voting will predictably fail in cases where a majority of unreliable users will vote on the same task. In the presence of historical data and multiple input from same users on different tasks, it is natural to assume that there are ways to analyze the workers' activity and derive more accurate answers by isolating systematic spammers or low quality workers. We study and propose methods that can improve results by inferring user's reliability taking into account all input.

Crowdsourcing algorithms have been the subject of research for several years. Recent benchmarks such as the one in (Sheshadri and Lease 2013) and (Hung et al. 2013) compare different approaches across different dimensions. Supervised approaches can benefit from access to golden data and/or significant prior knowledge on the crowd or tasks. Empirical results suggest that no single method is universally superior, as comparative performance varies with the domain and the dataset in question.

We focus here on a simple form of the problem, the unsupervised *binary crowdsourcing problem,* where workers answer Yes/No questions about items. An example of a binary crowdsourcing problem is to determine whether a Wikipedia edit is vandalism, or whether a given webpage is appropriate for children. One can view the problem as a process of probabilistic inference on a bipartite graph with workers and items as nodes, where both worker reputations and item labels are unknown. The problem of indirect inference of human reputation was first studied in the 70s, long before the advent of the internet and crowdsourcing marketplaces, with the description of the Expectation Maximization algorithm (Dawid and Skene 1979). Approaches closely related to EM were proposed in (Smyth et al. 1994) and (Raykar et al. 2010). A variational approach to the problem was recently proposed in (Karger, Oh, and Shah 2011) and a mean field method was proposed in(Liu, Peng, and Ihler 2012). The approach of (Karger, Oh, and Shah 2011), which we abbreviate by KOS, is proved to be asymptotically optimal as the number of workers tends to infinity, provided there is an unlimited supply of statistically independent workers. Nevertheless, we will show that KOS is not optimal in extracting the best estimate of the underlying truth from a finite amount of work performed.

In this paper, we begin by describing a general framework for binary inference, in which a beta-shaped belief function is iteratively updated. We show that the KOS approach corresponds to a particular choice of update for the belief functions. Casting the KOS algorithm as a belief-function update enables us to gain insights on its limitation. In particular, we show that the KOS approach is not optimal whenever the amount of work performed is non-uniform across workers, a very common case in practice, as well as whenever there is correlation between the answers provided by different workers. Furthermore, in cases involving a finite number of workers and items, correlation is created simply by iterating the inference step too many times; indeed, we show that the performance of KOS generally gets worse with increasing number of inference iterations.

We describe two variations of the beta-shaped belief function update, which we call the *harmonic* and the *parameter estimation* algorithms. The harmonic update is a simple update equation that aims at limiting the effects on the final result of worker correlation, worker supply finiteness, and difference in amount of work performed by different workers. There is no deep theoretical underpinning for the harmonic approach, and its virtues are its simplicity and empyrical robustness. The parameter estimation approach, in contrast, consists in updating the belief beta-distributions by estimating, at each iteration, the beta-distribution parameters that provide the best approximation for the true posterior belief distribution after one step of inference. We develop in detail the parameter estimation procedure, showing that it is feasible even for large practical crowdsourcing problems.

For the purpose of this study, we model the user reputations by a one-coin parameter. However, the ideas we describe for performing the update user and item distributions are extensible to a more complex two-coined model, such as the two-coin extension of (Dawid and Skene 1979) and (Liu, Peng, and Ihler 2012), where it is assumed that users can have different true positive and true negative rates. While our empirical study focuses on unsupervised settings, supervised variants with our methods are possible, as the methods maintain distributions on both user reputations and item qualities, and we can use knowledge on the crowd or the items to impose informative priors on the distributions.

We evaluate the harmonic and parameter estimation approaches both on synthetic data, and on large real-world examples. On synthetic data, we show that for non-regular graphs and for correlated responses, both our approaches perform well, providing superior performance compared with the KOS and EM methods. We then consider four real-world datasets generated by other research groups via Amazon Mechanical Turk. One dataset, kindly provided by the author of (Potthast 2010), consists of Wikipedia edits classified by workers according to whether they are vandalism; other two datasets contain annotations by non-experts on questions of textual entailment and temporal ordering of natural language texts (Snow et al. 2008), and a fourth dataset comes from the Duchenne experiment (Whitehill et al. 2009). The parameter estimation approach shows statistically significant superiority against other methods with respect to average recall for two of the real-word datasets, while it ties with other methods in the other two cases. The harmonic approach provides performance closely approaching that of parameter estimation, while maintaining the advantage of simplicity and ease of implementation.

Overall, the experiments show that the harmonic and parameter estimation approaches provide robust approaches to binary crowdsourcing that improve on the state of the art in a wide variety of settings.

Definitions

We consider a set U of users and I of items. A *task* consists in a user $u \in U$ giving an answer $a_{ui} \in \{+1, -1\}$ on item $i \in I$; by convention, $+1$ denotes a positive answer (i.e. 'true', or 'yes') and -1 a negative answer. We denote the answer set by A. The set of users, items and answers form a bipartite graph $E \subseteq U \times I$, whose edges represent the users' votes on the items. We call the users that have voted on an item $i \in I$ the neighborhood of i, denoted by $\partial i = \{u \mid (u, i) \in E\}$. Likewise, the neighborhood ∂u of a user u consists of the set $\{i \in I \mid (u, i) \in E\}$ of items that u voted on. The goal of the binary crowdsourcing problem consists in aggregating the user votes into item labels. One can view this as a double inference task: infer the most likely reliability of workers (which we can view as latent variables) based on their answers, and then use the worker's inferred reliabilities to infer the most likely labels of the items.

Viewing this as problem of probabilistic inference on a graphical model (the bipartite graph), its optimal solution is intractable (Koller and Friedman 2009). There exist several approaches to tackle this problem (Karger, Oh, and Shah 2011; Dawid and Skene 1979; Liu, Peng, and Ihler 2012; Raykar et al. 2010).

The KOS algorithm (Karger, Oh, and Shah 2011)

A recent approach to the binary crowdsourcing problem is described in (Karger, Oh, and Shah 2011). The approach is closely related to belief propagation (BP) (Yedidia, Freeman, and Weiss 2003), and executes on the bipartite voting graph in the style of BP by passing messages from workers to items and back in iterations. We give the pseudocode of KOS approach in Figure 1. The authors present results on synthetic graphs showing the superiority of this method to EM (Dawid and Skene 1979), and prove that the approach is asymptotically optimal for regular graphs, that is, as the size of the graph tends to infinity their approach is up to a constant factor as good as an oracle that knows the reliability of the workers.

Mean Field Approximation (Liu, Peng, and Ihler 2012)

Liu et al.(Liu, Peng, and Ihler 2012) propose a variation to the EM method (Dawid and Skene 1979). Making a beta distribution assumption for the probability q_j that a user j will provide a correct answer, the M-step update for their method is obtained using a variant written in terms of the posterior mean of the beta distribution rather than its posterior mode. The authors argue that this variation plays a role of Laplace smoothing. They report results that display a superior performance compared with EM and comparable performance to KOS (Karger, Oh, and Shah 2011) for some informative priors on users' quality. They also explore different models for users' voting. Instead of assuming a fixed reliability of users, they examine a two-coin model where each user has a varying reliability based on the task (specificity). Alternative models for user behavior have been considered and appear applicable in tasks requiring expertise such as the Bluebird Dataset (Welinder et al. 2010).

Beta belief distributions for users and items

We now show that the KOS algorithm, and our algorithms, can be interpreted as update rules for beta distributions of be-

```
Input  : bipartite graph $E$, answers $a_{ui}, k_{max}$
Output: Estimation of correct solutions $s_i \in \{+1, -1\}$
         for all $i \in I$.
foreach $(u, i) \in E$ do
    Initialize $y_{u \to i}$ with $Z_{i,j} \sim N(1, 1)$;
for $k = 1..., k_{max}$ do
    foreach $(u, i) \in E$ do
        $x^{(k)}_{i \to u} \leftarrow \sum_{u' \in \partial i \setminus u} a_{iu'} \cdot y^{(k-1)}_{u' \to i}$;
    foreach $(u, i) \in E$ do
        $y^{(k)}_{u \to i} \leftarrow \sum_{i' \in \partial u \setminus i} a_{i'u} \cdot x^{(k)}_{i' \to u}$;
foreach item $i$ do
    $x_i \leftarrow \sum_{u' \in \partial i} a_{iu'} \cdot y^{(k_{max}-1)}_{u' \to i}$;
Return estimate $\hat{s}_i = [sgn(x_i)]$ for all $i \in I$.;
```

Figure 1: KOS algorithm for labeling items using binary crowdsourced answers

lief on item value and user quality. This will provide an unifying framework to understand the properties of KOS and of our algorithms. Note that our setting of beta belief updates is not a variant of belief propagation; despite of the use of the term 'belief' in both cases, we maintain real-valued distributions of users reputations and item qualities, unlike belief propagation.

Similarly to (Liu, Peng, and Ihler 2012), we characterize users and items with probability distributions in the domain $[0, 1]$. The distribution of a worker represents the information on the worker's reputation or reliability, while the distribution of an item represents the information on its quality. The smaller the standard deviation of the distribution, the "peakier" the distribution, and the smaller the uncertainty over item quality or worker reliability. The higher the mean of the distributions, and higher the expected quality of workers or the expected truth value of items. A worker of perfect reliability has distribution $p(r) = \delta(r - 1)$ and a perfectly unreliable worker has distribution $u(r) = \delta(r)$, where δ is the Dirac delta function. A natural choice for the distributions over worker reliability and item quality is the *beta distribution.* A beta distribution $\text{Beta}(\alpha, \beta)$ with parameters α, β represents the posterior probability distribution over the bias x of a coin of which we saw $\alpha - 1$ heads (positive votes for items, truthful acts for users) and $\beta - 1$ tails (negative votes for items, false acts for users), starting from the uniform prior. An item whose distribution has $\alpha > \beta$ will have distribution median greater than 0.5, and be classified as true at the end of the inference process; conversely if $\alpha < \beta$.

A beta-distribution interpretation of KOS

The presentation of the KOS algorithm is slightly complicated by the fact that the algorithm, when computing the feedback to item i from users in ∂i, avoids considering the effect of i itself on those users. This leads to the message-passing presentation of the method that we see in Figure 1. If we allow the consideration of the effect of i on ∂i, we obtain a simpler version of KOS that *"allows self-influence"*. Such a version can be described succintly as follows. For every user u, initialize its reputation via $r_u \sim N(1, 1)$. Then, iteratively perform the updates:

$$r_i = \sum_{u \in \partial i} r_u \, a_{ui} \qquad r_u = \sum_{i \in \partial u} r_i \, a_{ui} \; . \tag{1}$$

Note that, at a step, the influence of user u on item i is that the amount $r_u \, a_{ui}$ is added to r_i (and similarly in the other direction, from items to users). After the desired number of iterations, decide the value of i by $\hat{s}_i = \text{sign}(r_i)$, for all $i \in I$.

We can view this algorithm as an update rule for beta distributions as follows. Every user u is associated with a beta distribution $\text{Beta}(\alpha_u, \beta_u)$ representing their truthfulness, and every item i is associated with the distribution $\text{Beta}(\alpha_i, \beta_i)$ representing its quality. Our interpretation maintains the invariants $r_u = \alpha_u - \beta_u$ and $r_i = \alpha_i - \beta_i$. Initially, we set $\alpha_u = 1 + r_u, \beta_u = 1$ for every $u \in U$, where r_u is initialized from the normal distribution as before. To perform the update step of (1), for each $i \in I$ we initialize $\alpha_i = \beta_i = 1$, and for each $u \in \partial i$, we increment α_i, β_i as follows:

$$\begin{cases} \alpha_i := \alpha_i + \alpha_u, & \beta_i := \beta_i + \beta_u \quad if a_{ui} > 0, \\ \alpha_i := \alpha_i + \beta_u, & \beta_i := \beta_i + \alpha_u \quad otherwise. \end{cases} \tag{2}$$

A similar update is then performed for each $u \in U$. It is easy to prove by induction that the above beta-distribution based algorithm, and the simplified (1) algorithms, are equivalent. We can obtain an analogous reformulation of the original KOS algorithm 1 by sending (α, β) pairs as messages, in place of single quantities x and y, exchanging α and β whenever $a_{ui} < 0$.

Limitations of KOS approach

The above re-statement of the KOS algorithm in terms of beta distribution updates sheds light on some of the limitation of the KOS algorithm.

Non-regular graphs. In real world scenarios, it is rare that items and workers are connected in a regular graph. Usually, some workers are very active providing multiple reviews, while others may provide only a few. Similarly, items have different popularity or visibility, some of them receiving many reviews, while others receiving only a few. In many real cases, power-law phenomena are in place.

The KOS algorithm may not perform well on non-regular graphs. To understand the reason, note that as the number of iteration progresses, the values of x, y (or α and β in our restatement) grow with a geometric rate related to the degrees of the nodes. Consider an item i, which is connected in the bipartite graph to two users, u_1 and u_2. The user u_1 is part of a large-degree subgraph; the user u_2 is instead part of a subgraph where all nodes (items and users) have small degree. Assume that u_1 and u_2 both give the same judgement (say, $+1$) about i. If the algorithm determines that u_1 has high reputation ($\alpha_{u_1} \gg \beta_{u_1}$), this reputation will be reflected in a strong certainty that the value of i is $+1$ ($\alpha_i \gg \beta_i$). In the subsequent iteration from items to users, we will have that the full amount of α_i will be added to α_{u_2}: the certainty of u_1 being truthful will be transferred to u_2. But this is of course inappropriate: u_2's vote on i is only one instance of agreement with the highly-reputed user u_2, and we should infer only a limited amount of certainty from one instance of behavior. In general, if the bipartite review graph is non-regular, the KOS algorithms will weigh excessively the evidence from the higher-degree portions of the graph. Our simulation results on artificial graphs will show that both the harmonic and the parameter estimation algorithms we propose outperform KOS on non-regular graphs.

Source dependence, and iterations over a finite graph. The additive nature of the KOS update rule makes the algorithm highly sensitive to the assumption of independent sources, and independence can fail, for two reasons.

First, the original sources (the users) are usually not statistically independent. For example, to answer a question such as "what is the phone number of this restaurant", most workers will consult a limited number of sources such as Google Maps, Bing, and Yelp, and choose the source they trust the most in case of conflict. The workers would not be performing statistically independent inferences on the phone number; rather, they would be influenced by their a-priory trust in the information sources. The issue of crowds deriving their information from a limited number of information sources has been studies in finance; (Hong, Page, and Riolo 2012) show that the resulting correlation can hinder the ability of groups to make accurate aggregate predictions.

Furthermore, and even more relevant to our context, statistical dependence is generated simply by running the KOS algorithm on a finite graph for many iterations. Indeed, if the graph has degree m, after n iterations we would need $(m-1)^n$ distinct sources for them to independently contribute to the value at a node. This is analogous to the fact that most of our ancestors $n > 30$ generations ago appear in multiple nodes of our genealogical trees, since there were several orders of magnitude fewer than 2^n humans at that time. In essence, for each item, the infinite tree of inference with branching $m-1$ is being folded inside the finite graph, and correlated information (corresponding to the same actual nodes in the graph) is being treated as if coming from independent graph nodes. The upshot is that after the first few initial rounds, the updates recycle the same information. Indeed, our experiments show that on finite graphs, the performance of KOS and other iterative methods peaks after a few initial rounds, and gets worse as the method reaches the fixed point. These empirical results appear to contradict the optimality results given in (Karger, Oh, and Shah 2011), but the contradiction is only apparent. The optimality results proved in (Karger, Oh, and Shah 2011) concern the behavior when the number of reviewers, and the size of the graph, grow to infinity; they do not concern the problem of optimally extracting information from a finite graph.

Our proposed algorithms are also affected by source correlation. However, our empirical results indicate that they are less affected than KOS. Intuitively, this is because our updates are performed based on reputation mean, rather than adding up the shape parameters.

Two proposed algorithms: Harmonic and Parameter-Estimation

We now describe two methods for the binary crowdsourced labels aggregation problem. Both algorithms model the distributions of item quality and user reliability via beta distributions, updating the distributions in iterative manner.

The *Regularized Harmonic Algorithm* is derived from the beta-distribution interpretation of KOS by adopting an update rule based on distribution means, rather than addition of shape parameters. This leads to a simple and efficient algorithm that performs well in presence of correlated information.

The *Beta Shape Parameter Estimation Algorithm* uses beta distributions to represent both item and worker distributions, and performs updates by first performing a Bayesian update, and then using parameter estimation to approximate the posterior distributions with beta distributions.

In both algorithms, we assume that each item is associated with a quality or ambiguity y that corresponds to a Bernoulli trial probability of the item being perceived as true by a perfectly reliable user. Similarly, each user has a probability x of telling the truth (i.e. report accurately the result of the bernulli trial of the item), and $1-x$ of lying (i.e., reporting the opposite of the observed result). We assume that y and x follow distributions that can be approximated by beta distributions.

The root reason why these algorithms outperform EM is that unlike EM, the algorithms explicitly represent (via the variance of the beta distributions) the amount of information we have on each user and item, so that they can distinguish users with the same averge quality but different amounts of certainty over it.

The Regularized Harmonic Algorithm

The *Regularized Harmonic Algorithm* represents the knowledge about a user u via a beta distribution $\mathrm{Beta}(\alpha_u, \beta_u)$, and the knowledge about an item i via a beta distribution $\mathrm{Beta}(\alpha_i, \beta_i)$. The update rule (2) adds the shape parameters α_u, β_u of users $u \in \partial i$ to compute the shape parameters of item i. Thus, a user u whose distribution has shapre parameters α_u, β_u has an influence proportional to $\alpha_u + \beta_u$. As α_u and β_u grow during the iterations, this can affect the performance over non-regular graphs, and in presence of correlated information, as discussed earlier. In the harmonic algorithm, the influence of a user is proportional to $|2p_u - 1|$, where $p_u = \alpha_u/(\alpha_u + \beta_u)$ is the mean of the beta distribution; and symmetrically for items. This leads to a more stable update rule, where differences in graph degree, and information correlation, have a more moderate effect on the final result. The detailed algorithm is given in Figure 2, where we use the standard notation $x^+ = (x + |x|)/2$ for the positive part of x.

Beta Shape Parameter Estimation Algorithm

The Beta Shape Parameter Estimation Algorithm, which we abbreviate by BSP, also models user and item distributions

```
Input : Bipartite graph $E \subseteq U \times I$, answers $a_{ui}, k_{max}$
Output: Estimation of correct solutions $s_i$ for all $i \in I$.
foreach user $u$ and item $i$ do
    $\alpha_u = 1 + \Delta$ for some $\Delta > 0$, and
    $\beta_u = \alpha_i = \beta_i = 1$.
for $k = 1..., k_{max}$ do
    foreach user $u \in U$ do $p_u \leftarrow \alpha_u/(\alpha_u + \beta_u)$
    foreach item $i \in I$ do
        $\alpha_i \leftarrow 1 + \sum_{u \in \partial i} (a_{ui}(2p_u - 1))^+$
        $\beta_i \leftarrow 1 + \sum_{u \in \partial i} (-a_{ui}(2p_u - 1))^+$
    foreach item $i \in I$ do $p_i \leftarrow \alpha_i/(\alpha_i + \beta_i)$
    foreach user $u \in U$ do
        $\alpha_u \leftarrow 1 + \sum_{i \in \partial u} (a_{ui}(2p_i - 1))^+$
        $\beta_u \leftarrow 1 + \sum_{i \in \partial u} (-a_{ui}(2p_i - 1))^+$
Return estimate vector $\hat{s}_i = \text{sign}(\alpha_i - \beta_i)$ for all $i \in I$.
```

Figure 2: Regularized Harmonic Algorithm

as beta distributions. The algorithm updates iteratively these distributions by first performing a pure Bayesian update, obtaining general posterior distributions, and then by re-approximating these poserior distributions by beta distributions having the same mean and variance. The idea of making a specific assumption about a distribution and performing approximate bayesian updates using parameter estimation is fairly classical; it was applied in a crowdwourcing context in (Glickman 1999) to the problem of computing chess and tennis player rankings from match outcomes making a normal distribution assumption for the strengths of the players.

In practice, BSP never computes the actual posterior distributions; these distributions are simply used as a mathematical device to derive update rules that are expressed directly in terms of shape parameters. To derive the update, assume that user u voted True (or Yes) for item i. We assume there is a prior for the quality of the item, given by distribution Beta(α_i, β_i). We can observe the event of a True vote by u on i when one of two mutually exclusive events occur: either the item was seen as true and the user reported it as true, or the item was seen as false, but the user flipped the observation. The probability of the former event is $x \cdot y$, and the probability of the latter is $(1 - x) \cdot (1 - y)$; given that the two events are mutually exclusive, the overall probability of a vote True is their sum. A Bayesian update for the item distribution after the user True vote yields:

$$g^{(k+1)}(y) \propto g^{(k)}(y) \int_0^1 (x \cdot y + (1 - x) \cdot (1 - y)) \cdot x^{\alpha_u - 1} \cdot (1 - x)^{\beta_u - 1} dx \tag{3}$$

BSP starts by assigning a prior to the users reputations. The choice of the prior is open. In most cases, we use a prior where users are considered weakly truthful, corresponding to a beta distribution with shape parameters $\alpha = 1 + \Delta$ and $\beta = 1$, where $\Delta > 0$ is small.

We use the votes of the users to update the items distribution by calculating the bayesian update through integration and normalization of Formula 3. The derived function is not a beta distribution and further updates in an iterative manner are intractable. We thus approximate the derived distribution by beta distribution through calculation of the expectation and variance of the derived distribution and estimation of the shape parameters of the beta distribution that corresponds to this expectation and variance. For details see our technical report (de Alfaro, Polychronopoulos, and Shavlovsky 2015). The procedure proceeds in a symmetrical way to update the user distributions from the item distribution and the votes. BSP performs these iterative updates of user and item distributions, either until the distributions converge (the difference across iterations is below a specified threshold), or until we reach a desired number of iterations. After the final iteration, we label with True (or Yes) the items i for which $\alpha_i \geq \beta_i$, and we label with False (or No) the others.

Excluding self-votes Similarly to the KOS method (Karger, Oh, and Shah 2011), for each user we can have separate distributions g_{ui} for all the items i the user voted on, where the distribution g_{ui} represents the information on u derived without using i directly; similarly for the distributions r_{ui} for items. The final estimate of item labels is obtained by adding all the differences $\alpha - \beta$ of the shape parameters of all the distributions for the item, and obtaining the sign. We provide a pseudocode for the method in Figure 3.

```
Input : bipartite graph $E \subseteq U \times I$, answer set $A$, $k_{max}$
Output: Estimation of correct solutions $s : \hat{s}(\{a_{ui}\})$
foreach $(u, i) \in E$ do
    Initialize user distributions $r_{ui}(x)$ with
    Beta$(1 + \Delta, 1)$ for some $\Delta > 0$;
    Initialize item distributions $g_{ui}(y)$ with Beta$(1, 1)$;
for $k = 1..., k_{max}$ do
    foreach item $i \in I$ do
        foreach item $u \in \partial i$ do
            Obtain $g_{ui}^{(k+1)}(y)$ through pure bayesian
            derivation;
            Obtain $\alpha_{ui}^i$ and $\beta_{ui}^i$ for $g_{ui}$ through shape
            parameter estimation;
            $g_{ui}^{(k+1)}(y) \leftarrow$ Beta$(\alpha_{ui}^i, \beta_{ui}^i)$;
    foreach user $u \in U$ do
        foreach item $i \in \partial u$ do
            Obtain $r_{ui}^{(k+1)}(y)$ through pure bayesian
            derivation;
            Obtain $\alpha_{ui}^u$ and $\beta_{ui}^u$ for $r_{ui}$ through shape
            parameter estimation;
            $r_{ui}^{(k+1)}(y) \leftarrow$ Beta$(\alpha_{ui}^u, \beta_{ui}^u)$;
Return vector $\hat{s}_i = \text{sign}(\sum_{u \in \partial i} \alpha_{ui}^i - \beta_{ui}^i)$ for all $i \in I$.
```

Figure 3: Beta Shape Parameter estimate (BSP) algorithm

Experimental study on synthetic data

We conducted experiments on synthetic regular graphs as in (Karger, Oh, and Shah 2011) and (Liu, Peng, and Ihler 2012). We also tested graphs whose vertex degrees follow a uniform distribution in a predefined range, and graphs whose vertex degrees follow a Pareto distribution. This is true in many graphs where task allocation is not defined by the algorithm: for example, popular items on Yelp will have much more votes than others, following some form of power law.

Independent users model In the spammer-hammer model of (Karger, Oh, and Shah 2011), users are either spammers or honest. Spammers provide random answers, that is, they reply True or False with 50% probability regardless of the true label, while honest workers report the truth. We also use a model where user accuracies follow a uniform distribution in $[0.5, 1]$, and a model where user accuracies follow a beta distribution with $\alpha = 0.03$, $\beta = 0.01$ which corresponds to a mean accuracy of 0.75 and variance ≈ 0.18. Parameter q represents the percentage of honest workers. We report the fraction of misclassified labels, averaged for 10 runs of the algorithm (on newly constructed graphs) which is an estimate of the probability of misclassification for a given method. For sets with balanced classes, the average error is a reliable performance measure. We also conduct experiments with varying class balance skew and report the F-1 measure which is more appropriate in this case.

The limited sources model In one set of experiments, we induce correlation to users' answers by forcing them to seek their answers by a limited set of sources. We fix the number of sources to 5, the sources vote on the items with a predefined accuracy, then users pick one of the sources to seek the answer in the following manner: the most popular source is picked with a given probability, the second most probable source is picked with 20% less probability and likewise for the remaining sources. This depicts a realistic scenario where some internet sources are more popular than others or contain more information. The correlation induced in this manner can be a challenge for algorithms that aim to obtain reliable aggregations from crowdsourcing. We could get overconfident in an answer which is in fact only a replicated erroneous answer from a single source.

Algorithms We implement the BSP approach, the Harmonic approach, EM (Dawid and Skene 1979), KOS approach (Karger, Oh, and Shah 2011), and AMF-EM (Liu, Peng, and Ihler 2012). The two last methods are the state-of-the-art methods of the literature. We also implemented a variation of the KOS method suitable for non-regular graphs (uniform and Pareto) which we call 'Regularized KOS algorithm'. It works in the same manner but regularizes item messages by the square root of the total votes they received (for and against). In that way, it prevents items becoming very highly reputed simply because they have received more votes than average. In all synthetic cases, we have a prior on user's reputation that is slightly truthful with $\Delta = 0.001$, that is, we are almost agnostic with respect to human workers reliability.

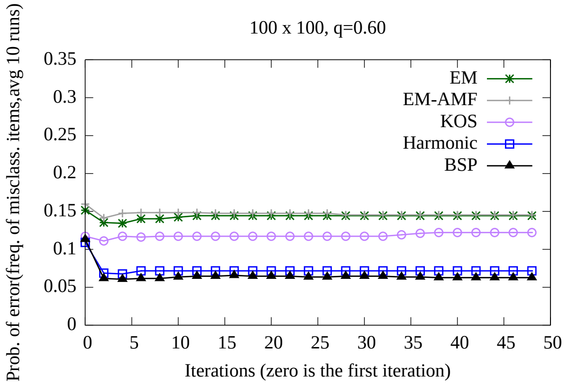

Figure 4: Results for a 100X100 regular bipartite graph with 5 votes per item, q=0.6

Statistical significance testing We do not know the distribution of the average error or other performance measures of the algorithms. However, the result of independent runs across different random graphs are i.i.d. random variables. For large samples (≥ 30), due to the central limit theorem, the arithmetic mean of a sufficiently large number of iterates of independent random variables is approximately normally distributed, regardless of the underlying distribution, and the z-test is an appropriate statistical hypothesis test method in this case (Sprinthall 2011). We conducted the z-test for large samples (≥ 80) across different runs, using the unbiased sample variance to approximate the distribution variance, to confirm the relevance of results that we see in the plots. When we report a superior result, we have confirmed its statistical significance with a high enough sample size that makes the test reject the null hypothesis with very low critical values ($<< 0.01$), i.e. very high confidence.

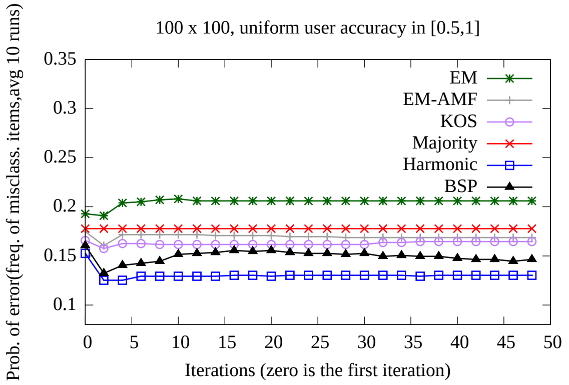

Figure 5: Results for a 100X100 regular bipartite graph with uniformly distributed user accuracies

Results Figure 4 shows the results for a regular bipartite graph of 100 users and 100 items for levels of spamming at $40\%(q = 0.6)$ using the spammer-hammer model. Figure 5 shows results for uniformly distributed user accuracies, and Figure 6 shows resuts using the beta distribution to model user accuracies. We observe the superiority of the BSP and Harmonic approaches, confirmed through significance testing, to other techniques while we note that simple majority voting outperforms EM in Figure 5, and both EM and EM-AMF in Figure 6. We omit results on majority voting which is outperformed by all iterative methods (the plot already contains several methods).

Figure 7 shows the comparative performance using the limited sources model. Again, the BSP method and the harmonic approach demonstrate the best performance, yet, the harmonic approach is the one performing better. The BSP method observably deteriorates as the iterations increase, reinforcing errors due to the independence assumption which does not hold.

Figure 8 shows the performance for a graph where the number of votes on each item follow a Pareto distribution with shape parameter 0.9. We used both KOS standard and regularized approaches. EM-AMF, BSP and harmonic outperform the rest with a slight lead of BSP. KOS approach performs poorly, the regularized variant we implemented provided benefits but the performance is still low. We also confirm that the KOS method diverges from its best performance as iterations increase when the regularity assumption is broken.

Finally, Figures 9 and 10 show the performance for a graph that is 1000×1000, an order of magnitude higher in the number of nodes for the beta distribution accuracies and limited sources models. We confirm the superiority of both the BSP and Harmonic methods in Figure 9 and that of the BSP method in Figure 10.

We omit results for graphs with uniform distribution of edges which demonstrate a similar performance.

Figure 11 shows the performance with increasing votes per item. Unsurprisingly, majority voting steadily improves with increasing number of votes (as the law of big numbers materializes and the percentage of inaccurate responses converges to their probability). KOS significantly improves with increasing number of votes and matches BSP and Harmonic in absolute correctness for 13 and 15 votes per item. Surprisingly, 9 votes per item are worse than 7 votes per item for all three iterative methods (and we confirmed its statistical significance). The reasons for this are not evident. An explanation is that while 9 and 11 votes are more than 7, a graph with 9 and 11 votes per item has a wider span and an error may propagate more widely compared to a graph with fewer votes. For the case of 7, the graph hits a sweet spot where the votes are high enough that it improves over fewer votes, but also few enough so the impact of errors remains localized. A formal explanation of the phenomenon may be an object of further study.

Figure 12 shows the F1 measure as we vary the skew of the balance of classes. BSP and Harmonic (alternately) have the lead in performance as skew varies, with Harmonic demonstrating less sensitivity across the variance of the skew.

Experimental study on real-world data

To demonstrate the relevance of our approach on real-world data, we conducted experiments with datasets obtained through Amazon's Mechanical Turk. For the workers priors, we make the assumption they are 4 times more likely to be reliable than not, that is, $\Delta = 3$.

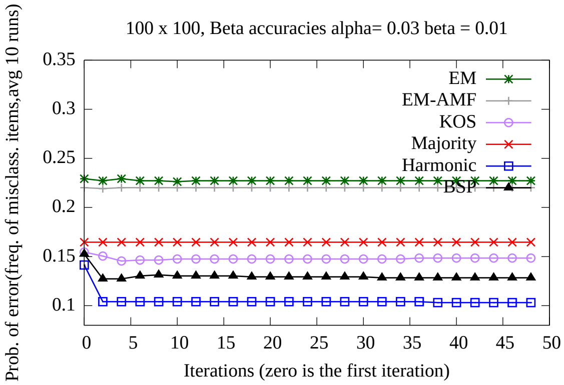

Figure 6: Results for a 100X100 regular bipartite graph with Beta distributed user accuracies

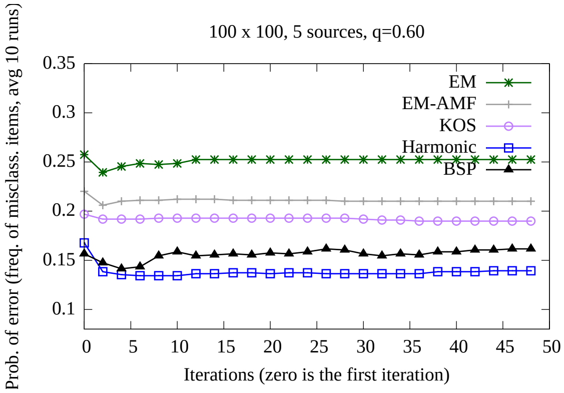

Figure 7: Results for a 100X100 regular bipartite graph with the limited sources model (5 sources), q=0.6

Statistical significance testing We sample the edges of the bipartite graph that corresponds to the real world dataset with a probability of 90%, constructing a set of random subgraphs. For each of the random subgraphs we obtain the required performance using two different methods measure and run the z-test on the results to determine statistical significance of difference in performance between the methods.

Wikipedia edits Mechanical Turk data

We tested our method on a real world dataset involving judgements of Mechanical Turk workers on a set of Wikipedia edits. The question that the workers addressed was whether a particular edit was valid or the result of vandalism. The set of workers and edits are therefore an instance of the binary crowdsourcing problem that our methods address. For this dataset, we do not have an explicit ground truth for the Wikipedia edits. We obtain the ground truth for a set of edits through the redundant nature of the data. For a particular set of edits we have a very high number of votes. We isolate the edits that have more than 25 votes and obtain

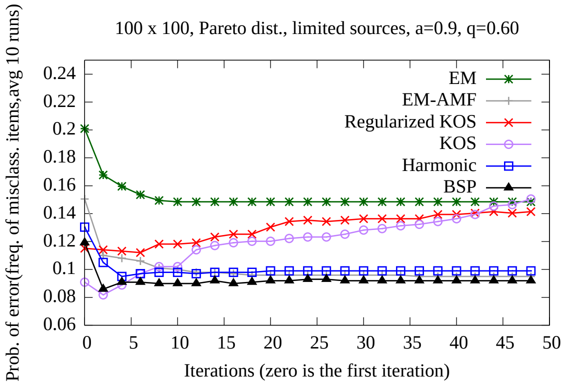

Figure 8: Results for a 100X100 Pareto distribution graph with the limited sources model (5 sources), q=0.6

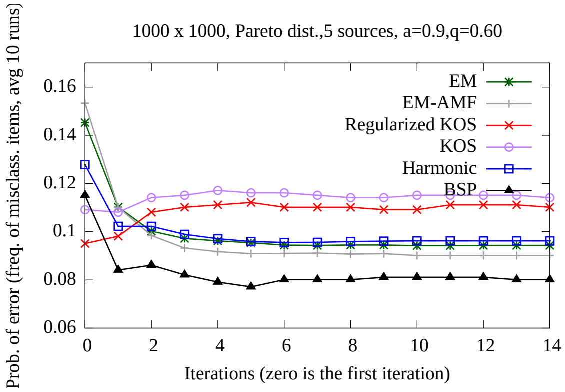

Figure 10: Results for a 1000X1000 Pareto distribution graph with the limited sources model (5 sources), q=0.6

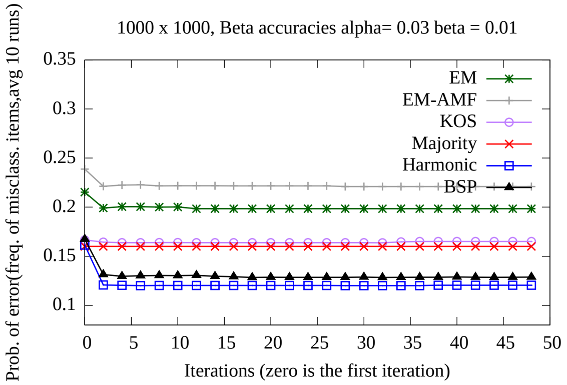

Figure 9: Results for a 1000X1000 regular bipartite graph with Beta distributed user accuracies

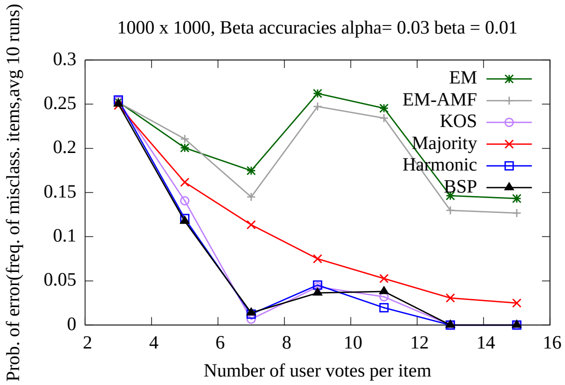

Figure 11: Comparative performance with increasing number of votes per item

what we deem as ground truth through majority voting. The number of votes is high enough that we can with high confidence consider that the ground truth for those edits can be obtained only through the consensus.

The idea is that majority voting approaches correctness as the number of votes increase. This is the same intuition used in (Piech et al. 2013) to measure the precision of crowdsourced grading algorithms. While we acknowledge that a small margin of error might remain after averaging 25 judgements, we argue that the residual error is likely to be small. Sadly, obtaining futher information on what has been reverted on the Wikipedia is not practical and also not reliable. Edits can be reverted multiple times, and reversions do not always indicate bad content leading to disaffected editors (Adler, de Alfaro, and Pye 2010). Also, training ML models with independent set of parameters to detect vandalism converges to the golden set obtained through annotators (Mola-Velasco 2012), an additional indication that obtaining labels through highly redundant annotation from human workers is a reliable method.

To construct the bipartite graph, we obtain the earlier 5 votes for all the highly-voted items. These votes, and the corresponding workers and edits, together with the ground truth that we obtain by using all the votes, define the bipartite graph on which we run and test our algorithms. We end up with a graph of 1927 edits and 677 workers. We omit the iterations performance study, which is available at the technical report (de Alfaro, Polychronopoulos, and Shavlovsky 2015), due to space limitations. A summary of the iterations study, all approaches start with similar or close results to majority voting, and then the error rates go higher as the iterations increase, apparently due to reinforcing wrong beliefs on some users reputation. BSP appears to be robust to this effect, and is the only method that has a lead over simple majority voting reducing the error rate of the vandalism classification. The difference with majority voting is small (in the order of 1%) but statistically significant.

NLP Mechanical Turk datasets

We also conducted experiments with publicly available NLP datasets obtained through Amazon's Mechanical Turk (Snow et al. 2008). Unlike the Wikipedia dataset, for these datasets we have a given ground truth. Similarly to (Liu, Peng, and Ihler 2012), we report results for increasing number of votes per task. We show the performance of the methods after 5 iterations. The results of igure 13 for the textual entailment dataset (RTE) confirm the divergence of

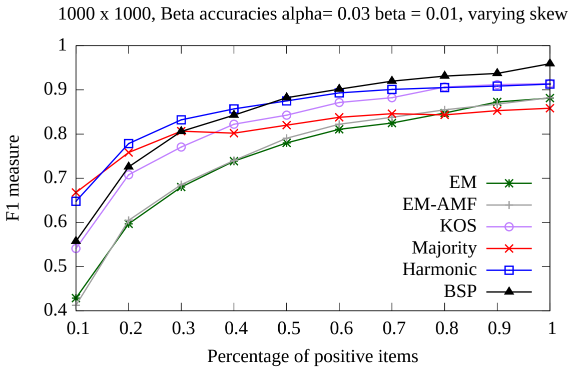

Figure 12: Comparative performance with varying skew of the two classes

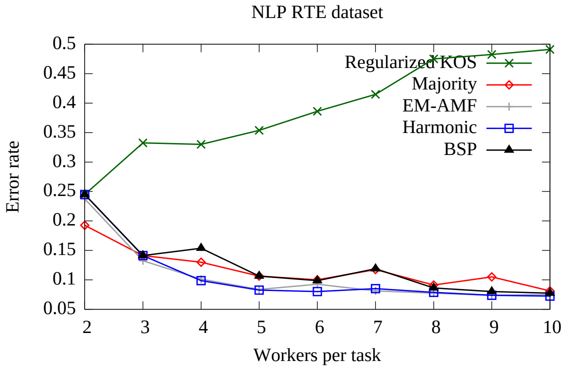

Figure 13: Comparative performance for the RTE dataset

the (Karger, Oh, and Shah 2011) approach and show that with the exception of KOS which performs bad, the iterative methods have generally a comparable performance. We omit the vanilla EM from the already dense plot as its performance is comparable. Due to space limitations, we omit the results of the NLP Temp dataset (available in the technical report) as they show a qualitatively similar performance.

The Duchenne experiment dataset

The last real-world dataset we examined is the publicly available dataset from the Duchenne experiment which was used in (Whitehill et al. 2009), and contains binary answers from MTurk users on whether a face is smiling along with the ground truth. We omit the comparative results for the average classification error which is available at the technical report. For the average error, regularized KOS is again diverging, while EM-AMF emerges as slightly superior to majority voting for the average error, also confirmed through significance testing.

Average recall for real-world datasets

The results on the real-world datasets are inconclusive with respect to which method is superior, in line with other studies (Sheshadri and Lease 2013) that show performance of methods oscillating across datasets. Average recall has emerged as an efficient evaluation measure where the data has a skewed class distribution. F-1 measure is often an appropriate measure but it is not symmetric with respect to positive and negative classes and the real-world datasets we tested have variable skew (Wikipedia: 1460 positives/ 467 negatives, RTE 400 negatives/positives, Temp 203 negatives/259 positives, Duchenne 101 negatives/58 positives) where the negative class is not always a minority. We thus report the average recall along the datasets in Table 1 for all methods.

	Wiki	RTE	Temp	Duchenne
Reg. KOS	0.720	0.506	0.534	0.611
Harmonic	0.782	0.929	0.937	0.731
BSP	**0.810**	0.923	**0.947**	**0.752**
Majority	**0.810**	0.919	0.937	0.743
EM-AMF	0.762	0.925	0.938	0.468
EM	0.731	0.925	0.934	0.427

Table 1: Comparative average recall for real-world datasets. Statistically significant superiority is marked with bold. We report performance at the fixed point.

Though both majority voting and BSP are superior to other methods for the average recall of the Wikipedia dataset, we cannot reject the null hypothesis for the performance difference between them, that is, it is not statistically significant for the size of the samples we use. BSP has superior performance for the NLP Temp and Duchenne experiment datasets, whereas we note that the EM methods obtain a particularly low average recall, even though they emerge as slighly superior for the average error measure which is blind to the classes, which they achieve by labeling items with the label of the majority class in an indiscriminate way.

The reason for the low performance of the iterative methods in some of the real-world datasets is likely to be due to the underlying user model. We use a one-coin model to describe user behavior. In some cases, users do not behave according to the simple Bernoulli assumption. A two-coin model is likely to have benefits on the performance. Also, for the purposes of the study we do not assume any information on the class skew and we have non-informative priors on the items. Our methods can support informative priors, Indeed, modifying the item prior to reflect the class skew provides benefits. Estimating the data skew is feasible in real-world settings by sampling, so we expect our iterative methods to adapt well in a non unsupervised setting where we use existing knowledge on the users and items to impose informative priors.

Conclusion

We describe two new methods for aggregating binary crowdsourcing tasks. The first, named *harmonic,* is derived heuristically, and it is simple to describe and implement. The second method has a principled derivation, rooten in the iterative update of estimates of user reliability and item quality distributions via bayesian parameter estimation. Our experimental evaluation indicates that the two algorithms perform better on many occasions, both on synthetic data, and on real-world data. The algorithm based on bayesian parameter estimation exhibits a slightly superior performance. In many practical applications, however, the simplicity, robustness and ease of implementation of the harmonic approach may make it the preferable approach. The practice in the literature is to report results on the fixed point, that is, when iterations stop producing changes. Our experiments, by following performance through iterations, show that the methods can achieve their best performance well before the fixed point, only to occasionally worsen as the fixed point is achieved. We attribute this to the recycling of information after several iterations that induces artificial correlation. We discovered that a good heuristic criterion is to perform $k = \frac{\log(\max\{m,n\})}{log(2\cdot d)}$ iterations, where m, n are the numbers of items of users and items, and d is the geometric average of all node degrees. This helps to propagate all available information across the length of the graph, while keeping correlation low. Stopping after k iterations instead of converging to the fixed point offers benefits for all iterative methods.

Acknowledgement This research was partially supported by the NSF Award DUE-1432690.

References

Adler, B.; de Alfaro, L.; and Pye, I. 2010. Detecting wikipedia vandalism using wikitrust. *Notebook papers of CLEF* 1:22–23.

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

de Alfaro, L.; Polychronopoulos, V.; and Shavlovsky, M. 2015. Reliable aggregation of boolean crowdsourced tasks. Technical Report UCSC-SOE-15-16, University of California, Santa Cruz.

Glickman, M. E. 1999. Parameter estimation in large dynamic paired comparison experiments. *Applied Statistics* 48:377–394.

Hong, L.; Page, S. E.; and Riolo, M. 2012. Incentives, information, and emergent collective accuracy. *Managerial and Decision Economics* 33(5-6):323–334.

Hung, N. Q. V.; Tam, N. T.; Tran, L. N.; and Aberer, K. 2013. An evaluation of aggregation techniques in crowdsourcing. In *Web Information Systems Engineering–WISE 2013*. Springer. 1–15.

Karger, D. R.; Oh, S.; and Shah, D. 2011. Iterative learning for reliable crowdsourcing systems. In *NIPS*, 1953–1961.

Koller, D., and Friedman, N. 2009. *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press.

Liu, Q.; Peng, J.; and Ihler, A. T. 2012. Variational inference for crowdsourcing. In *NIPS'12*, 701–709.

Mola-Velasco, S. M. 2012. Wikipedia vandalism detection through machine learning: Feature review and new proposals: Lab report for pan at clef 2010. *arXiv preprint arXiv:1210.5560*.

Piech, C.; Huang, J.; Chen, Z.; Do, C.; Ng, A.; and Koller, D. 2013. Tuned models of peer assessment in moocs. *arXiv preprint arXiv:1307.2579*.

Potthast, M. 2010. Crowdsourcing a Wikipedia Vandalism Corpus. In Crestani, F.; Marchand-Maillet, S.; Chen, H.-H.; Efthimiadis, E.; and Savoy, J., eds., *33rd International ACM Conference on Research and Development in Information Retrieval (SIGIR 10)*, 789–790. ACM.

Raykar, V. C.; Yu, S.; Zhao, L. H.; Valadez, G. H.; Florin, C.; Bogoni, L.; and Moy, L. 2010. Learning from crowds. *J. Mach. Learn. Res.* 11:1297–1322.

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

Smyth, P.; Fayyad, U. M.; Burl, M. C.; Perona, P.; and Baldi, P. 1994. Inferring ground truth from subjective labelling of venus images. In Tesauro, G.; Touretzky, D. S.; and Leen, T. K., eds., *NIPS*, 1085–1092. MIT Press.

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 *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, 254–263. Stroudsburg, PA, USA: Association for Computational Linguistics.

Sprinthall, R. C. 2011. *Basic statistical analysis*.

Welinder, P.; Branson, S.; Belongie, S.; and Perona, P. 2010. The Multidimensional Wisdom of Crowds. In Lafferty, J.; Williams, C. K. I.; Shawe-Taylor, J.; Zemel, R. S.; and Culotta, A., eds., *NIPS*. 2424–2432.

Whitehill, J.; Wu, T.-f.; Bergsma, J.; Movellan, J. R.; and Ruvolo, P. L. 2009. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *NIPS*, 2035–2043.

Yedidia, J. S.; Freeman, W. T.; and Weiss, Y. 2003. Exploring artificial intelligence in the new millennium. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. chapter Understanding Belief Propagation and Its Generalizations, 239–269.