Who Moderates the Moderators? Crowdsourcing Abuse Detection in User-Generated Content*

Arpita Ghosh
Yahoo! Research
Santa Clara, CA, USA
arpita@yahoo-inc.com

Satyen Kale
Yahoo! Research
Santa Clara, CA, USA
skale@yahoo-inc.com

Preston McAfee
Yahoo! Research
Burbank, CA, USA
mcafee@yahoo-inc.com

Abstract

A large fraction of user-generated content on the Web, such as posts or comments on popular online forums, consists of abuse or spam. Due to the volume of contributions on popular sites, a few trusted moderators cannot identify all such abusive content, so viewer ratings of contributions must be used for moderation. But not all viewers who rate content are trustworthy and accurate. What is a principled approach to assigning trust and aggregating user ratings, in order to accurately identify abusive content?

In this paper, we introduce a framework to address the problem of moderating online content using crowdsourced ratings. Our framework encompasses users who are untrustworthy or inaccurate to an unknown extent — that is, both the content and the raters are of unknown quality. With no knowledge whatsoever about the raters, it is impossible to do better than a random estimate. We present efficient algorithms to accurately detect abuse that only require knowledge about the *identity* of a *single* 'good' agent, who rates contributions accurately more than half the time. We prove that our algorithm can infer the quality of contributions with error that rapidly drops as the number of observations increases; we also numerically demonstrate that the algorithm has very high accuracy for much fewer observations. Finally, we analyze the robustness of our algorithms to manipulation by adversarial or strategic raters, an important issue in moderating online content, and quantify how the performance of the algorithm degrades with the number of manipulating agents.

1 Introduction

The Web is growing, and the extent of active user participation on the Web is growing with it as well. Most websites that display content now also allow user comments and discussions, and attract a huge volume of user posts— for instance, the popular website Slashdot receives over thousands of comments on its posts each day, while the more popular articles on Yahoo! News might receive thousands of comments *each*. Many of these posts are insightful or informative, or otherwise add value to the reader's experience, but there is a large quantity of abuse as well: according to a recently released study, as much as 95% of user-generated posts on Web sites are spam or malicious[1].

As the volume of user-generated content (UGC) increases, a solitary trusted moderator cannot single-handedly deal with the problem of identifying bad content. Even a small group of hand-picked moderators is grossly inadequate— quoting from Slashdot, "The (moderation) system worked well, but as Slashdot continued to grow, it was obvious that these 25 (trusted) people wouldn't be enough to keep up with the thousands of posts we were getting each day". The only solution to dealing with the problem of moderating the huge volume of user-generated content is to crowdsource the job of moderating content as well. Indeed,

*This work was published in the 12th ACM Conference on Electronic Commerce (EC), 2011. There was a calculation error in the paper published in the proceedings of EC 2011 in the computation of the error rate. This version fixes that error.

[1] http://www.daniweb.com/news/story258407.html

most websites now allow users the option to rate individual pieces of content as good or bad, usually using thumbs-up/thumbs-down style buttons, with a view to using this feedback to infer quality.

The problem of moderating user-generated content using viewer ratings, though, is not an easy one — *if* users were all reliable, these ratings would be adequate to determine whether a particular contribution is inappropriate. But *not all users who rate content rate it 'right'*: even the most evidently undesirable posts receive a non-zero number, and sometimes a significant fraction, of thumbs-up votes, and conversely perfectly good content is voted down by trolls as well. To make matters worse, raters are not simply 'good' or 'bad', consistently rating content right or consistently rating it wrong— the same user might rate some contributions accurately but some others inaccurately, either deliberately or due to random error from misjudging the threshold of acceptability. So we have a situation where in addition to content itself being good or bad, the *ratings* themselves may be questionable, leading to the following dilemma: when a user votes down a piece of content, how do we know whether the *rating or the content is bad*?

If we only observe users' ratings for a single contribution, the best we can do in the absence of any other information is to treat all ratings as equally likely to be correct, and use simple aggregation by majority vote. But there is typically more information available, since most systems that allow users to rate content also require them to be logged in to do so— this allows us to observe users' ratings over many contributions. Now suppose we have some way to estimate the appropriateness of some contributions, perhaps using ratings from some 'good' users whose identities are known to the system. Can these extra observations be used to do better, and under what conditions— what and how much prior information do we need about raters, and how reliable do we need our good raters to be, to be able to effectively use this information?

Most moderation schemes used by websites hosting user-generated content indeed use information about a user's past behavior to decide whether she is reliable or trustworthy enough to moderate, and how much moderation power to assign to her[2]. But what is a measure of reliability or trust anyway? We emphasize that the primary goal of a moderation system is *not* deciding which raters are trustworthy or to what extent, but rather only to accurately infer the quality of each contribution— any estimate of a rater's quality is merely a means towards this end. The question of deciding what a suitable measure of trust is and how to estimate it, is therefore closely related to the question of how to *aggregate* ratings to accurately estimate the appropriateness of each contribution. What is a principled approach to assigning 'moderation power' and aggregating ratings, and how do we quantify susceptibility to manipulation?

Our Contributions. In this paper, we introduce a framework within which to address the problem of *moderating* user-generated content, when the viewers who rate content are not perfectly accurate or trustworthy, *and* have unknown accuracy. Our model allows us to quantify the robustness of the moderation scheme in the presence of manipulation by adversarial raters of different kinds: this is an important factor in the UGC moderation context where trolling, either to promote a particular agenda or to generally attack the moderation system, is a common occurrence.

While we phrase our discussion in terms of abuse on online forums, e.g., spam or malicious content in comments on Slashdot or YouTube, our framework also applies to other instances of UGC where some kinds of contributions are clearly undesirable, and recognizable (albeit with error) as such. One particular example is *biased* reviews of products or services, such as when a user trying to promote a particular product or service (or bring down a competing product or service) posts an evidently biased review; this is not uncommon, for example, on Amazon. We also note that while in this paper we only address the problem of moderation, or identifying inappropriate content, this model could provide a first step for addressing further questions like accurately inferring more fine-grained ratings or accurately ranking content as well; see §7.

In our model, contributions are either 'good' or 'bad', and each rater, or agent, i rates contributions correctly with some *unknown* probability ψ_i: ψ_i can be thought of as a continuous version of the measure of a rater's trustworthiness that is used for assigning moderation privileges by various online moderation schemes. Clearly, with no knowledge at all about agents' rating abilities ψ_i, it is impossible to predict contributions' qualities more accurately than random guessing. We show that all we need to know is the

[2]A number of these (e.g., StackOverflow) assign points based on behavior, and assign moderation privileges to different extents based on the number of accumulated points.

identity of a single agent with $\psi_i > 1/2$, *i.e.*, an agent who rates accurately more than half the time, to achieve good performance— we emphasize that we do not need to know the exact value of ψ for this agent, nor do we need her to rate well almost all the time, or rate all items. We present efficient algorithms based on spectral decomposition that, armed with the identity of this one 'good' agent, infer the quality of contributions with error rate that rapidly falls as we observe more and more contributions; we also numerically verify that the algorithm identifies inappropriate content with high accuracy even for much fewer observations.

Organization. We begin with presenting our model in §2. In §3, we give an algorithm for estimating the quality of contributions based on relating the top eigenvector of the matrix $\mathbb{E}[UU^\top]$, where U is the matrix of observations, to the vector of actual contribution qualities, and analyze its performance (§3.1). In §4, we present and analyze an online version of this algorithm which estimates the rating abilities of the agents, ψ_i, using the same eigenvalue decomposition, and then infers the quality of each additional contribution as it arrives based on these estimates of ψ_i and its ratings. (We note that these algorithms have almost indistinguishable performance in numerical simulations, see §6.) Finally, in §5, we analyze the robustness of our algorithms to manipulation designed to attack the entire moderation system, *i.e.*, affect all ratings, as well as to manipulations targeted at a single contribution, such as by a group of agents with an agenda.

Related Work. The problem of moderating user-generated content is a major issue that is widely discussed on online forums, and there are even companies[3] consulting on best practices for addressing UGC abuse. These focus on solutions like content filtering based on the text of contributions, and ad hoc solutions for the problem of aggregating viewer ratings. While there has recently been work [13, 15, 8, 14] on inferring labels obtained from crowdsourcing image-annotation tasks, the specifics of the problem and consequently the model and approaches differ from ours, and to the best of our knowledge there is no work on formal models or algorithmic approaches to the problem of crowdsourcing the moderation of UGC.

There is a growing body of work on user-generated content. The recent work [6] addresses quality in UGC from the perspective of incentivizing users to contribute high-quality content. While [6] assumes *nonstrategic accurate ratings* by viewers and uses these to incentivize contributors, we do not address the issue of disincentivizing spammers — we merely take their presence as given and ask how to accurately identify spam content in the presence of inaccurate ratings. We also note that while we address robustness to manipulation, we do not address the issue of actually incentivizing users to rate accurately. [3] proposes an algorithm to incentivize users to rate honestly in *revenue-generating* ranking and reputation systems like search engines or online retailers such as Amazon that is based on sharing revenue with raters; whether these incentives and models apply to UGC is an open question.

The problem of moderating online content is reminiscent of the literature on voting [4], as well as recommender [1] and reputation systems [5]. The important distinction with the voting and recommender systems literature arises because items, or candidates, have an *underlying quality* in our problem: a contribution is either abuse or not[4].In contrast, agents have different preferences over items in voting problems, *i.e.*, items have no underlying common quality. This distinction also applies to recommender systems, where different users might have different opinions about the same item; again, ratings for items are subjective and there is no notion of underlying item quality. Thus the goal of a recommender system is to decide whether a *user* might like a particular item given his and other users' ratings, whereas our problem is to identify whether the *item* is good or bad.

The distinction with the reputation system literature is the following: the goal of a moderation system is to judge content, and *not* the raters— any estimate of raters' abilities is useful only to the extent that it helps accurately rate content. Specifically, if a user is producing bad and good comments, we would like to identify the bad comments as bad and the good ones as good, rather than infer that the user produces

[3] www.eModeration.com

[4] We note that here we are not interested in distinguishing between excellent and mediocre/poor content, *i.e.*, in ranking content according to quality, but only in identifying abuse or spam which exist plentifully in online UGC sites.

good content half the time. While one could imagine estimating the 'reputation' of each contribution via user ratings, the nature of the problems addressed by the reputation system literature, such as building up a high reputation to cash in on it or starting afresh with a new identity after earning a bad reputation, do not quite apply to this mapping since contributions are not active agents in the sense used in reputation systems. We note that there is also work on inferring reputation in the presence of Sybil attacks, *i.e.*, the creation of numerous fake identities; see for instance [16]. Our algorithms can tolerate a small constant fraction of Sybil nodes.

Technically, the analysis of our algorithm is based on concentration bounds for random matrices. Similar analysis has been applied for data mining previously [2], where the main technical tool used is a high probability bound on the spectrum of random matrices. However, such techniques yield very weak bounds in our application. We resort to a different technique, a recently developed Chernoff-type concentration inequality [10] for sums of random rank one matrices, which give us much tighter performance guarantees. To our knowledge, this is the first application of such Chernoff bounds to the analysis of data mining algorithms.

2 Model

The main components of our model are the *contributions*, which are of unknown quality, and the *raters*, who rate these contributions with unknown accuracy: we would like to aggregate these ratings in such a way as to accurately identify the quality of contributions. While quality in UGC spans the whole spectrum from excellent contributions to mediocre-poor content which is neither harmful nor valuable, all the way down to abuse, we focus here on abuse, or unambiguously bad content such as spam, malicious links, posts on entirely unrelated topics, and explicit comments. We next formally describe the model.

There are n users, or agents, $i = 1, \ldots, n$, who rate content. There are T contributions, or items, $t = 1, \ldots, T$. Each item is either *good*, corresponding to having quality $q_t = 1$, or *bad*, corresponding to having quality $q_t = -1$. The 'bad' items correspond to spam or abusive contributions that are unambiguously undesirable to the website. The remaining 'good' contributions, while not necessarily all of equal quality, are the non-spam contributions.

The system does not know the values of q_t, and would like to infer the qualities of items using agents' ratings.

Agents rate items as ± 1, corresponding to voting using the thumbs-up/thumbs-down button which is very widely used online to obtain viewer feedback on content. Different agents have different *probabilities* of rating items correctly: ψ_i is the probability that agent i rates items correctly, *i.e.*, that her rating agrees with the true quality of the item. That is, suppose $u_{ti} \in \{-1, 1\}$ is agent i's rating for item t. Then,

$$u_{ti} = \begin{cases} q_t & \text{w.p. } \psi_i \\ -q_t & \text{w.p. } 1 - \psi_i. \end{cases}$$

We emphasize that the values of ψ_i are *unknown* to the system, *i.e.*, the ψ_i are latent variables and the system does not know in advance how well each agent rates contributions.

In general, agents might not rate all items. We will use p_i to denote the probability that agent i rates an item. For simplicity, we will present most of our analysis for $p_i = 1$, the case where all agents rate all items, and indicate how to extend the analysis to incomplete ratings.

With no knowledge whatsoever about the ψ_i values, it is easy to see that there is no hope of correctly inferring the quality of a contribution even if we have ratings from all agents on a very large number of items. We therefore make the following very weak assumption about the information the system has about the ψ_i. We assume that the agent knows the identity of *one* 'good' or trustworthy rater, where our requirement of goodness is also very weak— suppose agents are numbered so that this good agent is agent 1. We will assume that agent 1 rates items correctly strictly more than half the time: $\psi_1 > 1/2$.

We emphasize that we do not assume that the exact *value* of ψ_1 is known to the system, nor that ψ_1 is very close to 1— we simply need to know the *identity* of a single user with $\psi > 1/2$. We will sometimes use γ to denote how much better than random agent 1 rates, *i.e.*, $\gamma = \psi_1 - 1/2$; by assumption, $\gamma > 0$.

Definition 2.1 (Competence: $\kappa, \bar{\kappa}$)**.** *Define the* competence *of an agent i to be $\kappa_i = (2\psi_i - 1)^2 = 4(\psi_i - \frac{1}{2})^2$: κ_i is a number in $[0, 1]$ which indicates how informative i's rating is in determining the quality of an item, with 0 being the least informative corresponding to $\psi_i = 1/2$, and 1 being fully informative corresponding to $\psi_i = 1$ or $\psi_i = 0$. Define the total competence $\kappa = \sum_i \kappa_i$, and the average competence*

$$\bar{\kappa} = \frac{\kappa}{n} = \frac{4}{n} \sum_{i=1}^{n} (\psi_i - \tfrac{1}{2})^2.$$

The average competence will show up repeatedly in the analysis of our algorithm, with the error being inversely proportional to the average competence.

Given a set of observed ratings U and knowing that agent 1 has $\psi_1 > 1/2$, our goal is to infer the qualities of items, *i.e.*, the vector $q = [q_t]$, as well as possible.

Notation. q is the $T \times 1$ column vector of item qualities, and ψ is the $n \times 1$ vector of probabilities of agents voting correctly. For a vector v, we will use $\hat{v}$ to denote the unit vector in the direction of v, *i.e.*, $\hat{v} = \frac{1}{\|v\|} v$, where $\|v\|$ denotes the ℓ_2 norm of v.

We use $A^\top$ to denote the transpose of a matrix A. We use $\|A\|_2$ to denote the spectral norm of a matrix A, *i.e.*, its largest singular value.

3 Estimating q

In this section, we will present our first algorithm for estimating the qualities q using the observations U. The algorithm is based on a spectral decomposition of the matrix $UU^\top$, and also forms the basis of an online algorithm which we describe in the next section. We begin with some preliminaries that motivate the algorithm, and then analyze and bound the error rate as a function of the number of observations n, T and the average competence $\bar{\kappa}$ in Theorem 3.1. We discuss the efficiency of the eigenvector computation required by the algorithm in §3.3.

Let $U = [u_{ti}]$ represent the matrix of observed ratings, where the ith column u_i is the vector of agent i's ratings on the items $1, \ldots, T$. The expected value of u_{ti} is

$$\mathbb{E}[u_{ti}] = q_t \psi_i - q_t(1 - \psi_i) = q_t(2\psi_i - 1).$$

Therefore, we have

$$\mathbb{E}[U] = q(2\psi - \mathbf{1})^\top,$$

where recall that q is the $T \times 1$ column vector of item qualities, ψ is the $n \times 1$ vector of probabilities of agents voting correctly, and $\mathbf{1}$ denotes the $n \times 1$ column vector with all coordinates equal to 1. Note that $\mathbb{E}[U]$ is a rank-one matrix, with left singular vector proportional to q, and right singular vector proportional to $(2\psi - \mathbf{1})$. This singular value decomposition of $\mathbb{E}[U]$ suggests an algorithm for estimating q from the singular value decomposition of U. To analyze this algorithm, though, we will use the fact that the top left singular vector of U is identical to the top eigenvector of $UU^\top$, and relate the top eigenvector of $UU^\top$ to the top eigenvector of its expected value $\mathbb{E}[UU^\top]$. Next, we write $\mathbb{E}[UU^\top]$ in terms of q and ψ (note that $\mathbb{E}[UU^\top]$ is not the same as $\mathbb{E}[U]\mathbb{E}[U^\top]$).

Let $\phi_{ti} \in \{-1, 1\}$ be a random variable taking the value 1 if agent i rates item t correctly (which happens with probability ψ_i), and -1 otherwise (which happens with probability $1 - \psi_i$). Then, $u_{ti} = \phi_{ti} q_t$. (While ϕ_{ti} has the same distribution for each t, the realization of course might be distinct and thus the random variables need separates names.)

If $t \neq s$, then

$$\phi_{ti}\phi_{si} = \begin{cases} 1 & \text{w.p. } \psi_i^2 + (1 - \psi_i)^2 \\ -1 & \text{w.p. } 1 - \psi_i^2 - (1 - \psi_i)^2. \end{cases}$$

so

$$\begin{aligned} \mathbb{E}[\phi_{ti}\phi_{si}] &= \psi_i^2 + (1 - \psi_i)^2 - (1 - \psi_i^2 - (1 - \psi_i)^2) \\ &= (2\psi_i - 1)^2. \end{aligned}$$

If $t = s$, then $\phi_{ti}\phi_{si} = 1$, so $\mathbb{E}[\phi_{ti}\phi_{si}] = 1$ also.

The (s,t) element in $UU^\top$ is $\sum_{i=1}^n u_{si}u_{ti} = q_s q_t \sum_{i=1}^n \phi_{si}\phi_{ti}$, with expected value

$$\begin{aligned}\mathbb{E}[\textstyle\sum_{i=1}^n u_{si}u_{ti}] &= q_s q_t \mathbb{E}\left[\textstyle\sum_{i=1}^n \phi_{si}\phi_{ti}\right] \\ &= \begin{cases} \sum_{i=1}^n (2\psi_i - 1)^2 & \text{if } t \neq s \\ n & \text{if } t = s. \end{cases}\end{aligned}$$

Therefore, given the latent model ψ_i of agents' probabilities of rating items correctly, we can write the matrix $\mathbb{E}[UU^\top]$ as

$$\begin{aligned}\mathbb{E}[UU^\top] &= \left(\textstyle\sum_{i=1}^n (2\psi_i - 1)^2\right) qq^\top + \left(n - \textstyle\sum_{i=1}^n (2\psi_i - 1)^2\right) I \\ &= \kappa qq^\top + (n - \kappa) I, \end{aligned} \tag{1}$$

where κ is the total competence (Definition 2.1), and I is the $T \times T$ identity matrix.

This matrix $\mathbb{E}[UU^\top]$ therefore has a very simple spectral decomposition: it has top eigenvalue

$$\lambda_1 = \kappa \|q\|^2 + (n - \kappa) = \kappa T + (n - \kappa)$$

with corresponding eigenvector q. The remaining eigenvalues λ_i, for $i \geq 2$, are all equal, with value $n - \kappa$, and the corresponding eigenvectors lie in the orthogonal complement of the space spanned by q. This inspires the following algorithm for estimating q from the observations U.

Algorithm Spectral-Rating.

1. Compute the top eigenvector v of the matrix $UU^\top$ (scaled so $\|v\| = 1$), and let $s = sgn(v)$ (component-wise sign).
2. If $u_1 \cdot s \geq 0$, set $\sigma = 1$, else set $\sigma = -1$.
3. Output $q' = \sigma s$ as the estimated quality vector.

The first step computes the top eigenvector v of $UU^\top$, inspired by the fact that the top eigenvector of $\mathbb{E}[UU^\top]$ is q. Now note that if v is an eigenvector of $UU^\top$, then so is $-v$, but these correspond to exactly opposite estimates of the items' qualities. How do we decide which sign to choose for the eigenvector? The second step uses the fact that agent 1 rates items correctly more than half the time to choose the sign for the eigenvector based on correlation with agent 1's ratings. The third step simply estimates the quality of item i as the sign of the corresponding entry in σs.

Now, *if* we had access to the matrix $\mathbb{E}[UU^\top]$, extracting its top eigenvector would give us exactly the vector of actual qualities q. But of course, we only have access to U, and therefore $UU^\top$. How closely does the top eigenvector of $UU^\top$, which is a random matrix with mean $\mathbb{E}[UU^\top]$, resemble the top eigenvector q of $\mathbb{E}[UU^\top]$? The difference between these two eigenvectors will tell us how close q', our estimate of the quality derived from the top eigenvector of $UU^\top$, is to the actual quality vector q. We next analyze this difference.

3.1 Analysis

Our main result is Theorem 3.1, the proof of which is structured as follows. First, we adapt a matrix version of a Chernoff bound which tells us that the random matrix $UU^\top$ and its expectation $\mathbb{E}[UU^\top]$ are not very different, as measured by the spectral norm, with some probability. We use this bound to say that with the same probability, every pair of corresponding eigenvalues of $UU^\top$ and $\mathbb{E}[UU^\top]$ are close. We use both these results to bound the angle between the top eigenvector v of the observed matrix $UU^\top$, and q, which is the top eigenvector of the expectation $\mathbb{E}[UU^\top]$. Finally, we translate this bound on the angle between the two eigenvectors to the error in the algorithm, which is the number of coordinates in which q and v differ in sign. The probability with which this bound on the error holds is given by a union bound over the probability

with which the Chernoff bound holds, and the probability that we choose the correct sign for v based on the correlation with the ratings of the known good agent.

We begin with the following bound, obtained by tweaking Theorem 3.1 from [10], on the difference between a random matrix and its expectation (measured by the spectral norm)[5]:

Lemma 3.1 (Theorem 3.1 in [10])**.** *If vectors $y_1, y_2, \ldots, y_n \in \mathbb{R}^d$ are drawn from independent distributions such that for some parameters M and λ we have $\|y_i\| \leq M$ and $\|\mathbb{E}[\sum_{i=1}^n y_i y_i^\top]\| \leq \lambda$. Then there is an absolute constant c (independent of all other parameters) such that for any $\delta \in (0, \lambda)$ we have*

$$Pr\left[\left\|\sum_{i=1}^n y_i y_i^\top - \mathbb{E}\left[\sum_{i=1}^n y_i y_i^\top\right]\right\|_2 > \delta\right] \leq 2\exp\left(\frac{-c\delta^2}{\log(n)\lambda M^2}\right),$$

In the following text, c will refer to the absolute constant from the above theorem.

For our problem, the random vectors in question are $y_i = u_i$, where u_i is the $T \times 1$ column vector of ratings by agent i. Since each entry in u_i is either 1 or -1, $\|u_i\| = \sqrt{T}$, so $M = \sqrt{T}$.Furthermore, $\sum_i u_i u_i^\top = UU^\top$, and as calculated earlier, $\mathbb{E}[UU^\top] = \lambda_1 = \kappa T + n - \kappa$, so the parameter $\lambda = \lambda_1$. Therefore, for our problem, setting $y_i = u_i$, $M = \sqrt{T}$, and $\lambda_1 = \kappa T + n - \kappa$, we have, for any $\delta \in (0, \lambda_1)$,

$$Pr\left[\left\|UU^\top - \mathbb{E}[UU^\top]\right\|_2 > \delta\right] \leq 2\exp\left(\frac{-c\delta^2}{\log(n)\lambda_1 T}\right). \tag{2}$$

Now let $\eta \in (0, 1)$ be a confidence level parameter. Since the total competency κ can be expected to grow linearly with n, assume now that n is large enough so that $\kappa \geq c^{-1}\log(4/\eta)\log(n)$. Choosing $\delta = \sqrt{c^{-1}\log(n)\lambda_1 T \log(4/\eta)} \leq \lambda_1$, the RHS above becomes smaller than $\eta/2$. Thus, w.p. at least $1 - \eta/2$, we have

$$\|UU^\top - \mathbb{E}[UU^\top]\|_2 \leq \delta. \tag{3}$$

We assume this is the case from here on.

The following lemma (Appendix A), says that if the difference of two positive semidefinite matrices (here, $UU^\top$ and $\mathbb{E}[UU^\top]$) has a small spectral norm, then the top eigenvectors of the two matrices cannot be too far away from each other if (one of) the matrices have an adequate spectral gap, *i.e.*, difference between the first and second eigenvalue.

Lemma 3.2. *Let A and B be symmetric, positive semidefinite matrices such that $\|A - B\|_2 \leq \delta$. Let v_1 and w_1 be the unit eigenvectors corresponding to the top eigenvalues of A and B respectively. Let λ_1 and λ_2 be the top two eigenvalues of B. Then*

$$(v_1 \cdot w_1)^2 \geq 1 - \frac{\lambda_2 + 3\delta}{\lambda_1}.$$

This is indeed the case with the matrix $\mathbb{E}[UU^\top]$, where the difference between the top two eigenvalues is large when the number of items T is large, which is the regime of interest, corresponding to moderating a large number of contributions. Applying this lemma (with $A = UU^\top$ and $B = \mathbb{E}[UU^\top]$), we get that

$$(\hat{q} \cdot v)^2 \geq 1 - \frac{\lambda_2 + 3\delta}{\lambda_1},$$

where $\hat{q} = \frac{q}{\sqrt{T}}$ is the unit vector in the direction of q, and v is the unit eigenvector of $UU^\top$ computed in Step 1 of the algorithm.

Using the values $\lambda_1 = \kappa T + (n - \kappa)$ and $\lambda_2 = (n - \kappa)$, we get that

$$(\hat{q} \cdot v)^2 \geq 1 - \frac{n - \kappa + 3\delta}{\lambda_1}.$$

[5]Note that the statement of the bound in [10] assumes that the vectors y_i are drawn independently from identical distributions. It is easy to check that their analysis carries through to give the stated bound even if the distributions of the y_i vectors are different, as long as the vectors are drawn independently.

For our choice of $\delta = \sqrt{c^{-1}\log(n)\lambda_1 T\log(4/\eta)}$, we get that with probability at least $1-\eta/2$, we have

$$(\hat{q}\cdot v)^2 \geq 1-\epsilon/2, \tag{4}$$

where,

$$\epsilon = 2\cdot\frac{n-\kappa+3\sqrt{c^{-1}\log(n)\lambda_1 T\log(4/\eta)}}{\lambda_1} < 6\sqrt{\frac{\log(n)\log(4/\eta)}{c\bar{\kappa}n}}+\frac{2}{\bar{\kappa}T}. \tag{5}$$

So far we have shown that $(\hat{q}\cdot v)^2$ must be small with high probability. In Lemma 3.3 below, we show that when T (and n) are large enough, we choose the right sign for v: specifically, if $T > \frac{2}{\gamma^2}\log(\frac{4}{\eta})$, then with probability at least $1-\eta/2$, we have $\hat{q}\cdot(\sigma v)\geq 0$ (recall that $\gamma = \psi_1 - 1/2$). In this case, we can we conclude that $\hat{q}\cdot(\sigma v)\geq\sqrt{1-\epsilon/2} > 1-\epsilon/2$, *i.e.*, the angle between the vectors $\hat{q}$ and v is small with high probability.

Next we will translate this to a bound on the actual number of errors made by the algorithm in its estimate. Let $M = \{i : q_t = -q'_t\}$ denote the set of items on which our estimate is wrong. Since $\hat{q}$ and σv both have unit norm, we have

$$\|\hat{q}-\sigma v\|^2 = 1+1-2(\hat{q}\cdot\sigma v) < \epsilon. \tag{6}$$

But,

$$\|\hat{q}-\sigma v\|^2 \geq \sum_{t\in M}\left(\frac{q_t}{\sqrt{T}}-\sigma v_t\right)^2 \geq \sum_{i\in M}\frac{1}{T} = \frac{|M|}{T}. \tag{7}$$

The second inequality follows because for $t\in M$, $q_t\neq q'_t$, and $q'_t = sgn(\sigma v_t)$ so that $|\frac{q_t}{\sqrt{T}}-\sigma v_t|\geq\frac{1}{\sqrt{T}}$. This implies that the fraction of errors is bounded by ϵ.

The following lemma shows that we choose the right sign for the estimated ratings vector when T and n are large enough.

Lemma 3.3. *If $T > \frac{2}{\gamma^2}\log(4/\eta)$ and $\frac{n}{\log(n)} > \frac{128}{c\bar{\kappa}^2}$, then with probability at least $1-\eta/2$, we have $\hat{q}\cdot(\sigma v)\geq 0$.*

Proof. Assume without loss of generality that the vector v obtained at the end of step 1 of the algorithm satisfies $\hat{q}\cdot v\geq 0$. Then we want to show that with high probability, $u_1\cdot s\geq 0$, and thus we choose $\sigma = 1$ and $\hat{q}\cdot(\sigma v)\geq 0$.

For this, we make use of the fact that we know that user 1 is good with $\psi_1 > 0.5$. We have $\psi_1 = 0.5+\gamma$, where $\gamma > 0$ is a constant. Then for any item t, $\mathbb{E}[u_{t1}q_t] = \psi_1-(1-\psi_1) = 2\gamma$. Thus, $\mathbb{E}[u_1\cdot q] = 2\gamma T$. Hoeffding bounds say that for T independent random variables $X_1, X_2, \ldots, X_T\in\{-1,1\}$, for any $\delta > 0$, we have

$$\Pr\left[\mathbb{E}\left[\sum_{t=1}^{T}X_t\right]-\sum_{t=1}^{T}X_t\geq\delta\right]\leq\exp\left(\frac{-\delta^2}{2T}\right).$$

Applying this bound to our setting with $X_t = u_{t1}q_t$ and $\delta = \gamma T$, we get

$$\Pr[u_1\cdot q < \gamma T] < \exp(-\gamma^2 T/2).$$

So if $T > \frac{2}{\gamma^2}\log(4/\eta)$, then the probability that $u_1\cdot q < T\gamma$ is at most $\eta/2$. Thus, with probability at least $1-\eta/2$, we have $u_1\cdot q\geq\gamma T$, which implies that

$$|\{t : u_{1t} = q_t\}|\geq(1/2+\gamma/2)T.$$

Now, because we assumed that $\hat{q}\cdot v\geq 0$, the same calculations as in inequalities (6) and (7) applied to the vectors $\hat{q}$ and v (instead of σv) imply that

$$|\{t:\ q_t\neq s_t\}| < \epsilon T.$$

Thus, we have

$$|\{t:\ u_{1t} = s_t\}| > (1/2+\gamma/2-\epsilon)T\geq T/2,$$

which is equivalent to $u_1\cdot s\geq 0$. The inequality above holds if $\epsilon\leq\gamma/2$. This is indeed true for $T > \frac{2}{\gamma^2}\log(4/\eta)$, equation (5) implies that $\epsilon < \frac{1}{\bar{\kappa}}\sqrt{\frac{32\log(n)}{cn}}\cdot\gamma\leq\gamma/2$ for the specified range of n. □

This finally gives us the following high probability bound on the fraction of errors made.

Theorem 3.1. *Let $\eta \in (0,1)$ be a given confidence level parameter. Assume that $\kappa \geq c^{-1}\log(4/\eta)\log(n)$, $T > \frac{2}{\gamma^2}\log(4/\eta)$ and $\frac{n}{\log(n)} > \frac{128}{c\bar{\kappa}^2}$. Then with probability at least $1-\eta$, we have*

$$\frac{1}{T}|\{t:\ q'_t \neq q_t\}| \leq 6\sqrt{\frac{\log(n)\log(4/\eta)}{c\bar{\kappa}n}} + \frac{2}{\bar{\kappa}T}.$$

The theorem essentially says that when T and n are large enough, the fraction of errors is bounded with high probability by $O(\sqrt{\frac{\log(n)}{\bar{\kappa}n}} + \frac{1}{\bar{\kappa}T})$. Note also the inverse dependence of ϵ on the average competence, $\bar{\kappa}$: the larger the $\bar{\kappa}$, the smaller the error. This precisely quantifies how the agents' ability to rate accurately, *i.e.*, the ψ_i, affect performance. We discuss this in greater detail in §5.

We note that the algorithm actually delivers high accuracy even for much smaller values of n, T than predicted by this bound; we include numerical simulations demonstrating this in §6 (these simulations involve incomplete ratings, which are analyzed next.)

3.2 Accounting for Incomplete Ratings

In reality, not all agents rate all items, and a number of entries in the matrix U may be missing: the (i,j) entry in U is missing if agent j does not rate item i. The algorithm and analysis above can be extended to deal with this case as well, as follows. Suppose that agent i rates items independently with probability p_i. Set $u_{ti} = 0$ if agent i does not rate item t. Then,

$$u_{ti} = \begin{cases} q_t & \text{w.p. } p_i\psi_i \\ -q_t & \text{w.p. } p_i(1-\psi_i) \\ 0 & \text{w.p. } 1-p_i. \end{cases}$$

Using similar calculations as those for obtaining (1), it is easy to check the following:

$$\mathbb{E}[UU^\top] = (\textstyle\sum_i p_i^2(2\psi_i - 1)^2)qq^\top + (\sum_i p_i - p_i^2(2\psi_i-1)^2)I.$$

This expression reduces to (1) in the case that all $p_i = 1$.

We can now analyze the same algorithm as before. Define $\bar{\kappa}_p := \frac{\sum_i p_i^2(2\psi_i-1)^2}{\sum_i p_i}$: this is the analogue of $\bar{\kappa}_p$ in this setting. Repeating the same calculations as in the previous section, we immediately obtain the following bound on the error rate; the proof is deferred to Appendix B.

Theorem 3.2. *Let $\eta \in (0,1)$ be a given confidence level parameter. Assume that $\bar{\kappa}_p \sum_i p_i \geq c^{-1}\log(4/\eta)\log(n)$, $T > \frac{3}{p_1\gamma^2}\log(4/\eta)$ and $\frac{n}{\log(n)} > \frac{88p_1}{c\mathcal{S}^2}$. Then with probability at least $1-\eta$, we have*

$$\frac{1}{T}|\{t:\ q_t \neq q'_t\}| \leq 6\sqrt{\frac{\log(n)\log(4/\eta)}{c\bar{\kappa}_p\sum_i p_i}} + \frac{2}{\bar{\kappa}_p T}. \tag{8}$$

Structurally, the bound is similar to that of Theorem 3.1. The only difference is that the dependence on $\bar{\kappa}$ is replaced by a similar dependence on $\bar{\kappa}_p$. If all the p_i's are equal to some value r, then the $\bar{\kappa}_p = r\bar{\kappa}$; so the bound below says that the fraction of errors increases by a factor of $\frac{1}{r}$.

3.3 Computing Eigenvectors Efficiently

Our algorithm for estimating the qualities of contributions relies on computing the eigenvector of a $T \times T$ matrix. T is likely to be rather large in practice, since crowdsourcing moderation is necessary only when the number of contributions are large; also, the algorithm's performance improves with increasing T, so we would like to use as many observations as possible as input. Here we discuss the feasibility of this computation.

Eigenvector computations are extremely efficient in practice [11]. In particular, computing the top eigenvector of a positive semidefinite matrix can be done very efficiently using iterative methods such as the power method or the Lanczos iteration. The benefit of these iterative methods is that they only rely on matrix-vector products and thus can take advantage of sparsity in the matrix. In our case, even though the matrix $UU^\top$ can be dense, computing its product with a vector can be done in linear time in the number of non-zero entries of U: simply multiply by $U^\top$ first, and then multiply the result by U.

In our application, we actually do not even need to compute the top eigenvector exactly: rather, it suffices to approximate v, the top eigenvector of $UU^\top$, in the following sense. Find a vector y such that $(v \cdot \hat{y})^2 \geq (1 - \epsilon')$, where ϵ' is an error parameter. Then $\hat{y}$ can be used in place of v in Algorithm Spectral-Rating, and it can be shown easily that the fraction of errors only increases by $O(\epsilon')$. Thus, choosing $\epsilon' = \Theta(\epsilon^c)$ for some constant c, the performance of the algorithm remains essentially unchanged.

Now, because the difference between the top eigenvalues of $UU^\top$ is large, the power method converges extremely fast: in a constant number of iterations, *i.e.*, independent of T! This implies that a good approximation to the top eigenvector can be computed in linear time in the number of non-zero entries of U. The following lemma makes this precise (proof in Appendix B); the probability of success can be boosted to any desired level by repeating the power iteration with different random starting vectors and finally taking the one with the largest Rayleigh quotient with $UU^\top$.

Lemma 3.4. *Let $\epsilon' = \Theta(\epsilon^c)$ for some constant c. Let x be a randomly chosen vector in $\{-1, 1\}^n$. Let $y = (UU^\top)^k x$, Then for a large enough constant k, with probability at least $1/8$, we have $(v \cdot \hat{y}) \geq 1 - \epsilon'$.*

4 An Online Algorithm

In this section, we design an online version of the algorithm in the previous section for estimating the quality of each item, and provide bounds on its performance.

Algorithm Spectral-Rating can be thought of as a batch algorithm: it generates estimated ratings given a batch of data. Such a batch algorithm is not efficient enough to scale to large data sets, where a lightweight algorithm that uses some statistics computed from a reasonable batch of data to predict ratings of new items as they arrive, in an online fashion, is much preferable. We now present an online algorithm based on Algorithm Spectral-Rating, which first estimates the raters' competencies ψ_i using the ratings on some large number T of items, and then uses these estimated ψ_i and the observed ratings u_{ti} to estimate q_t for items $t \geq T + 1$. This algorithm is more efficient when an item's quality must be estimated immediately upon arrival, since we can simply use the precomputed ψ vector to output an estimate q_t, rather than re-computing an eigenvector of $UU^\top$ each time a new item arrives (*i.e.*, a new row is added to U).

It is well known that the optimal decision rule for estimating q_t, given the votes u_{ti}, is to use a weighted majority rule with logarithmic weights:

Lemma 4.1 ([9, 7])**.** *Given agents with probabilities ψ_i of voting correctly, the decision rule that maximizes the probability of returning the correct outcome is weighted majority with weights*

$$w_i = \frac{1}{2} \log \left(\frac{\psi_i}{1 - \psi_i} \right).$$

Of course, we do not know the ψ_i exactly but only have an estimate of the ψ_i from Spectral-Rating; we will use this approximate estimate of the ψ_i's as input to the weighted majority calculation.

The algorithm for estimating the ψ_i's is as follows:

Algorithm Estimate-ψ.

1. Run Algorithm Spectral-Rating to get the vector q'.
2. For each user i, output the estimate

$$\psi_i' = \frac{1}{2} \left(\frac{1}{T} (q' \cdot u_i) + 1 \right).$$

The prediction algorithm is essentially weighted majority with the estimated ψ_i. A small twist is a clipping of estimated ψ_i to the range $[\alpha, 1-\alpha]$ for some parameter α specified in the algorithm description to make the algorithm stable (note that the logarithmic weights tend to infinity at 0 or 1) and allow an error analysis.

Algorithm Predict

1. Run Algorithm Estimate-ψ to obtain ψ_i' and set $\psi_i'' = \min\{\max\{\psi_i', \alpha\}, 1-\alpha\}$, where $\alpha = 6\sqrt{\frac{\log(n)\log(4/\eta)}{c\bar{\kappa}n}} + \frac{2}{\bar{\kappa}T} + \sqrt{\frac{\log(4n/\eta)}{4T}}$.
2. Compute $w_i = \frac{1}{2}\log(\frac{\psi_i''}{1-\psi_i''})$.
3. When item $t \geq T+1$ arrives, output the estimate of its quality $q_t' = sgn\left(\sum_i w_i u_{t,i}\right)$.

4.1 Analysis

Since the algorithms are based on Algorithm Spectral-Rating, we assume throughout this section that the conditions for Theorem 3.1 hold, *i.e.*, $T > \frac{2}{\gamma^2}\log(4/\eta)$ and $\frac{n}{\log(n)} > \frac{128}{c\bar{\kappa}^2}$, so that with probability at least $1-\eta$, the error rate of Algorithm Spectral-Rating is bounded by $\epsilon = 6\sqrt{\frac{\log(n)\log(4/\eta)}{c\bar{\kappa}n}} + \frac{2}{\bar{\kappa}T}$.

We first analyze the quality of approximation to the ψ_i's achieved by Algorithm Estimate-ψ:

Theorem 4.1. *With probability at least $1-\eta$, we have for all i,*

$$|\psi_i'' - \psi_i| \leq \alpha.$$

Proof. First, note that $\mathbb{E}[q_i u_{ti}] = 2\psi_i - 1$. By the Hoeffding bounds, we have

$$\Pr\left[\left|\frac{1}{T}\sum_{t=1}^{T} q_t u_{ti} - (2\psi_i - 1)\right| > \epsilon'\right] < \eta/2n,$$

where $\epsilon' = \sqrt{\frac{\log(4n/\eta)}{T}}$. Next, by Theorem 3.1, we have

$$\Pr[|\{t:\ q_t \neq q_t'\}| > \epsilon T] < \eta/2,$$

where $\epsilon = 6\sqrt{\frac{\log(n)\log(4/\eta)}{c\bar{\kappa}n}} + \frac{2}{\bar{\kappa}T}$. This implies that with probability at least $1-\eta/2$,

$$\left|\frac{2}{T}\sum_{t=1}^{T} q_t u_{ti} - \frac{1}{T}\sum_{t=1}^{T} q_t' u_{ti}\right| < 2\epsilon,$$

since $|u_{ti}| = 1$. Applying a union bound over all i, we get that with probability at least $1-\eta$, for all i, we have

$$\left|\frac{1}{T}\sum_{t=1}^{T} q_t u_{ti} - (2\psi_i - 1)\right| < 2\epsilon + \epsilon' = 2\alpha.$$

This implies that

$$|\psi_i' - \psi_i| < 2\epsilon' = \alpha.$$

The clipping of ψ_i' to the range $[\alpha, 1-\alpha]$ to produce ψ_i'' maintains $|\psi_i'' - \psi_i| \leq \alpha$. □

We can now give bounds on the error rate of Algorithm Predict. This is expressed as a bound on the probability that an error is made on item q_{T+1}. Naturally, this bound also holds for any subsequent item $t > T+1$. The bound shows that as long as T is large enough so that $\alpha \ll \bar{\kappa}$ (note that $\alpha = O(\sqrt{\log(n)/T})$), the probability of making an error drops *exponentially* with n.

Theorem 4.2. *Assume that the assertion of Theorem 4.1 holds, i.e. for all i, we have $|\psi_i'' - \psi| \leq \alpha$. Assume further that n and T are large enough so that $\alpha \leq 0.05$. Then*

$$\Pr[q'_{T+1} \neq q_{T+1}] < \exp(-0.5(\bar{\kappa} - 24\alpha)n).$$

Proof. Without loss of generality, assume that $q_{T+1} = 1$. Algorithm Predict makes an error if $\sum_i w_i u_{T+1,i} < 0$. Consider the function $\exp(-\sum_i w_i u_{T+1,i})$. This is always non-negative, and at least 1 whenever Algorithm Predict makes an error. Hence, we have

$$\begin{aligned}
&\Pr[q'_{T+1} \neq q_{T+1}] = \mathbb{E}[1_{\sum_i w_i u_{T+1,i} < 0}] \\
&\leq \mathbb{E}[\exp(-\sum_i w_i u_{T+1,i})] \\
&= \prod_i \mathbb{E}[\exp(-w_i u_{T+1,i})] \quad \text{(by independence of } u_{T+1,i}) \\
&= \prod_i (\psi_i \exp(-w_i) + (1-\psi_i)\exp(w_i)) \\
&= \prod_i \left(\psi_i \cdot \sqrt{\frac{1-\psi_i''}{\psi_i''}} + (1-\psi_i) \cdot \sqrt{\frac{\psi''}{1-\psi_i''}} \right).
\end{aligned}$$

In Lemma 4.2 below, we bound the term in the product above by $\exp(-0.5(2\psi_i - 1)^2 + 12\alpha)$. Plugging in this bound, we get that

$$\Pr[q'_{T+1} \neq q_{T+1}] \leq \exp(-0.5(\bar{\kappa} - 24\alpha)n).$$

□

Lemma 4.2. *If $\alpha \leq 0.05$, the following bound holds:*

$$\left(\psi_i \cdot \sqrt{\frac{1-\psi_i''}{\psi_i''}} + (1-\psi_i) \cdot \sqrt{\frac{\psi''}{1-\psi_i''}} \right) \leq \exp(-0.5(2\psi_i - 1)^2 + 12\alpha).$$

The proof of this lemma is a technical calculation, and is deferred to Appendix B.

5 Manipulation

Finally, we address the issue of manipulation. We consider three kinds of agents: (i) agents who choose their value of ψ_i strategically to most degrade the performance of the algorithm; (ii) agents who deviate from the model and rate *each contribution adversarially* to most degrade the performance of the algorithm, *i.e.*, such agents' ratings need not be drawn according to a distribution given by ψ for any value of ψ, and (iii) agents who only want to influence the system's estimate of the quality of a *single* item — for instance, a group of agents might want to influence the outcome for a particular contribution either positively or negatively, but still rate other contributions accurately to the extent given by their ψ_i (either out of benevolence, or to maximize their 'trust' prior to manipulation). For all three cases, we analyze the robustness of the algorithm to manipulation, and quantify how the performance of the algorithm degrades with the number of manipulating agents.

Stochastic Manipulation. This first case, where agents continue to rate according to the model, but choose ψ to degrade the algorithm's performance, is easy. Recall that the performance of the algorithm depends on the competence of each agent in a simple way: the error rate ϵ is inversely proportional to the average competence $\bar{\kappa}$, *i.e.*, the smaller $\bar{\kappa}$, the larger the error. In fact, since this is the only term containing ψ_i in the expression for ϵ, a malicious agent who wants to attack the system by adversarially choosing ψ_i will do the most harm by choosing $\psi_i = 1/2$, *i.e.*, by simply randomly voting items good or bad. In particular, this algorithm is robust to agents who intentionally rate every (or most) items the opposite of its

actual quality, which corresponds to ψ_i close to zero: such agents in fact are quite useful to the algorithm since the algorithm can essentially invert their judgment. The performance bound follows by simply setting $\psi = 1/2$ for each manipulator in Theorem 3.1.

Adversarially choosing u_j. We now consider a different kind of malicious attack, where a malicious agent does not behave according to a probability distribution as postulated in the model at all, but chooses all her ratings adversarially to most degrade the performance of the algorithm. If there is more than one such agent, we will allow for the fact that these agents might collude, *i.e.*, the set of ratings from all these agents could be adversarially chosen.

Suppose there are b such bad agents, and the remaining $n_g = n - b$ agents rate stochastically. That is, the $n - b$ columns in U corresponding to these 'good' (though possibly not very competent) agents are stochastically generated, but the b columns corresponding to the bad agents are arbitrary vectors in $\{-1, 1\}^T$. In the following, all probability and expectation calculations are over the randomness of the good agents.

The bound below shows that the fraction of errors is essentially what can be obtained by applying Theorem 3.1 to the n_g good agents, plus an additional $\frac{6b}{\bar{\kappa}_g n_g}$ term. Thus, if $b \ll n$, and $\bar{\kappa}_g = \frac{\kappa_g}{n_g} > 0$, then this additional term becomes negligible. This means that Algorithm Spectral-Rating can tolerate a small constant fraction of malicious agents, even if these agents are colluding.

Lemma 5.1. *Under the assumptions of Theorem 3.1, we have that with probability* $1 - \eta$,

$$\frac{1}{T}|\{t : \ q_t \neq q'_t\}| \leq 6\sqrt{\frac{\log(n_g)\log(4/\eta)}{c\bar{\kappa}_g n_g}} + \frac{2}{\bar{\kappa}_g T} + \frac{6b}{\bar{\kappa}_g n_g}.$$

Proof. Let $\kappa_g = \sum_{i \notin B}(2\psi_i - 1)^2$ be the total competence of the good agents, and let U_g be the submatrix of U composed of the columns corresponding to the good agents. Then, a calculation similar to those in the proof of Theorem 3.1 one show that

$$\mathbb{E}[U_g U_g^\top] = \kappa_g q q^\top + (n_g - \kappa_g) I,$$

and with probability at least $1 - \eta$,

$$\|U_g U_g^\top - \mathbb{E}[U_g U_g^\top]\|_2 \leq \sqrt{c^{-1}\log(n_g)\lambda'_1 T \log(4/\eta)}$$

where $\lambda'_1 = \kappa_g T + (n_g - \kappa_g)$.

Now note that

$$\|\mathbb{E}[UU^\top] - \mathbb{E}[U_g U_g^\top]\|_2 = \|\sum_{j \in B} u_j u_j^\top\|_2 \leq bT.$$

Hence, with probability at least $1 - \eta$, we have

$$\|UU^\top - (\kappa_g q q^\top + (n_g - \kappa_g) I)\|_2 \leq \delta + bT.$$

Again as before, by Lemma A.1, we get that

$$(\hat{q} \cdot v)^2 \geq 1 - \frac{\lambda'_2 + 3\delta + 3bT}{\lambda'_1},$$

where $\lambda'_2 = (n_g - \kappa_g)$. For our choice of parameters, we get that $(\hat{q} \cdot v)^2 \geq 1 - \epsilon/2$, where

$$\epsilon < 6\sqrt{\frac{\log(n_g)\log(4/\eta)}{c\bar{\kappa}_g n_g}} + \frac{2}{\bar{\kappa}_g T} + \frac{6b}{\bar{\kappa}_g n_g},$$

where $\bar{\kappa}_g = \frac{\kappa_g}{n_g}$. Reasoning as before, ϵ gives a bound on the fraction of errors. □

Choosing u_{tj} to influence outcome for a particular item t. Now we consider what happens if some set B of b agents are interested in manipulating the outcome for a single item, although not necessarily in bringing down the entire moderation system as a whole. This is quite a realistic possibility, since a spammer with a particular agenda might be interested only in making sure that a post on that agenda is not accurately identified as spam[6]: in fact, it is to these agents' benefit to rate other items accurately to acquire as much of the system's trust as possible before attempting to manipulate the outcome for their chosen item.

To analyze this kind of manipulation, we will use the online version of our algorithm from §4, which allows us to address the error for a specific contribution rather than an average error over contributions. We will show that as long as b is small compared to κ_g, the total competence of the good agents, the probability of making an error still remains exponentially small. In the following theorem, $n_g = n - b$ denotes the number of good agents, and $\beta = \frac{1}{2}\log(\frac{1-\gamma}{\gamma})$ is an upper bound on the magnitude of the weights w_i produced by algorithm Estimate-ψ, where $\gamma = \max\{\alpha, \min_i\{1-\psi_i\}, \min_i\{\psi_i\}\}$. If every ψ_i is bounded away from 0 or 1, a reasonable assumption, β will be a moderately sized constant, because of the logarithm.

The bound below is similar in spirit to the bound of Lemma 5.1. It says that the error rate is what one might expect by applying Theorem 4.2, times an additional factor of $\exp(\beta b)$. Thus, if $b \ll n$, and $\bar{\kappa}_g > 0$, then the additional factor becomes negligible. The implication is that Algorithm Predict is resistant to manipulation by a small constant fraction of malicious agents, even if they are colluding.

Lemma 5.2. *Under the assumptions of Theorem 4.2, we have*

$$\Pr[q'_{T+1} \neq q_{T+1}] < \exp(-0.5(\bar{\kappa}_g - 24\alpha - \beta\tfrac{b}{n_g})n_g).$$

Proof. Assume as before that $q_{T+1} = 1$. The bad agents can do the most damage by choosing all their own ratings to be -1. We now want to bound the probability that the weighted majority rule makes an error, i.e. $\sum_i w_i u_{i,T+1} < 0$. This is equivalent to

$$\sum_{i \notin B} w_i u_{i,T+1} < \sum_{i \in B} w_i.$$

The RHS above is bounded by βb. We now bound the probability that $\sum_{i\notin B} w_i u_{i,T+1} < \beta b$. For this, we use the same technique as before:

$$\begin{aligned}
\Pr[\sum_{i\notin B} w_i u_{i,T+1} < \beta b] = \; & \mathbb{E}[1_{\sum_{i\notin B} w_i u_{i,T+1} < \beta b}] \\
& \leq \mathbb{E}[\exp(\beta b - \textstyle\sum_{i\notin B} w_i u_{i,T+1})] \\
& = \; \exp(\beta b) \prod_{i\notin B} \mathbb{E}[\exp(-w_i u_{i,T+1})].
\end{aligned}$$

The rest of the analysis is the same as before, and results in specified bound on the error rate. □

6 Simulations

The performance bounds proved in the previous sections guarantee that the fraction of errors goes to 0 as n and T grow. However, the bounds given are worst-case and might yield needlessly pessimistic requirements on how large n and T need to be to obtain small error. In this section, we numerically evaluate the performance of our algorithms for small values of n and T, and show that the accuracy of the algorithm is high even for these regimes. Note that there are no computational difficulties associated with handling larger numbers of observations[7]; here, we only want to demonstrate that a very large amount of data is not essential for accuracy.

[6] This comment on a YouTube video clearly falls in this category: "hey everyone we have a group on facebook but now we have a youtube channel so? go to our channel and subscribe to us!"; it actually had 10 thumbs-up votes.

[7] The online algorithm's performance is almost identical to that of the batch algorithm for the same parameters, so even larger amounts of data can be used to improve accuracy without sacrificing speed.

We imagine two scenarios, and plot how the error rate of the algorithm changes with increasing $\bar{\kappa}$, the average competence of the raters.:

1. A modestly sized website with a relatively small number of users and a modest number of contributions, with modest rating probabilities p_i. Specifically, we set $n = 100$ users, $T = 1000$ contributions, and the p_i's are randomly drawn uniformly from $[0, 0.3]$. This corresponds to collecting an average of 150 thumbs-up/down ratings by users, a reasonable number.

2. A large website with a large number of users and contributions, but smaller p_i. Here we set $n = 1000$ users, $T = 5000$ contributions, and the p_i's are randomly drawn uniformly from $[0, 0.1]$.

Competencies for the raters are randomly drawn from the normal distribution with mean $0.5+t$ and variance 0.01 clipped to the $[0, 1]$ interval. Here t is a parameter that is changed to obtain different values of $\bar{\kappa}$.

While we ran both Algorithm Spectral-Rating and Algorithms Estimate+Predict, for clarity we only show results for Algorithm Spectral-Rating since the performance of Algorithms Estimate+Predict was nearly identical. The error rate depicted in the graph is the 90% quantile obtained from 100 runs of the algorithm. We use the 90% quantile instead of the mean because there are sometimes a few outliers with large error causing small spikes in the mean (note that our bounds also only hold with probability $1 - \eta$); the 90% quantile curve is smoother and shows the trend of the error rate quite clearly, while still being a reasonable indicator of the performance.

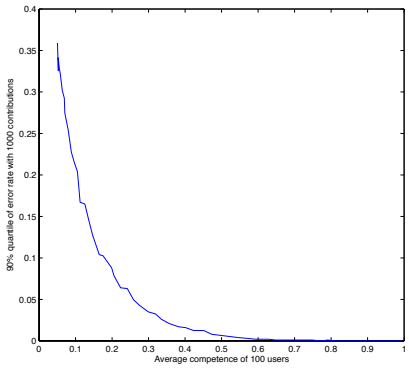

Figure 1: Average comptence v.s. 90% quantile of error rate, in Scenario 1: 100 users, 1000 contributions, p_i's drawn uniformly from $[0, 0.3]$.

The plots clearly demonstrate the effectiveness of the algorithm for much smaller values of T than required by the theoretical bounds. The performance improves rapidly as $\bar{\kappa}$ increases: when $\bar{\kappa}$ is zero, essentially all agents are rating randomly, and so the algorithm cannot do very much better than random guessing. However the error rates fall very rapidly to values in the range of a few percent even for very small values of $\bar{\kappa}$, and in the case of Scenario 2, the error rate improves even faster, falling to 0 for $\bar{\kappa} = 0.3$.

7 Discussion

In this paper, we introduced a model for the problem of moderating user-generated content using crowd-sourced ratings of unknown reliability, and presented efficient algorithms to accurately identify abusive content. There are a number of interesting directions for further work. First, while we focused on identifying unambiguously bad content, user ratings can also be used to infer the relative qualities of non-abusive contributions: not all non-spam UGC is of the same quality, and displaying better quality contributions more prominently improves user experience. How accurately can an algorithm infer the true rankings of the items

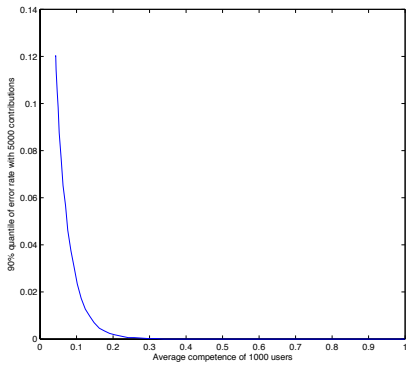

Figure 2: Average comptence v.s. 90% quantile of error rate, in scenario 2: 1000 users, 5000 contributions, p_i's drawn uniformly from $[0, 0.1]$.

according to their qualities in the presence of inaccurate ratings, and how sensitive is it to manipulation? For example, the order in which restaurants appear in the Yelp rankings likely strongly influences whether a user notices them or not; the ranking is generated precisely on the basis of user ratings that may not be accurate or honest. The model presented in this paper might provide a first step towards addressing this question.

Another interesting direction is robustness against manipulation: how can we design algorithms that are optimized against manipulation, to tolerate a larger number of malicious agents, and what are the trade-offs? Finally, we note that the most general model for UGC would involve both strategic contributors and strategic raters. The question of how to design a mechanism for this setting that both incentivizes contributors to produce high quality and raters to rate honestly is a promising direction for future work, potentially requiring insights from other fields to accurately model incentives for all agents.

References

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

[2] Y. Azar, A. Fiat, A. R. Karlin, F. McSherry, and J. Saia. Spectral analysis of data. In *STOC*, pages 619–626, 2001.

[3] R. Bhattacharjee and A. Goel. Algorithms and incentives for robust ranking. In *SODA*, pages 425–433, 2007.

[4] V. Conitzer. Making decisions based on the preferences of multiple agents. *Commun. ACM*, 53(3):84–94, 2010.

[5] E. Friedman, P. Resnick, and R. Sami. Manipulation-resistant reputation systems. *Algorithmic Game Theory*, 2007.

[6] A. Ghosh and R. P. McAfee. Incentivizing high-quality user generated content. In *Proc. WWW*, 2011.

[7] B. Grofman and L. Shapley. Optimizing group judgmental accuracy in the presence of interdependencies. *Public Choice*, 43:329–343, 1984.

[8] P. G. Ipeirotis, F. Provost, and J. Wang. Quality management on amazon mechanical turk. In *Proc. Workshop on Human Computation*, 2010.

[9] S. Nitzan and J. Paroush. Optimal Decision Rules in Uncertain Dichotomous Choice Situations. *International Economic Review*, 23(2):289–297, June 1982.

[10] M. Rudelson and R. Vershynin. Sampling from large matrices: An approach through geometric functional analysis. *J. ACM*, 54(4), 2007.

[11] L. N. Trefethen and D. Bau. *Numerical Linear Algebra*. SIAM, 1997.

[12] L. Trevisan. Computing eigenvectors. *http://lucatrevisan.wordpress.com/2011/01/29/cs359g-lecture-7-computing-eigenvectors/*.

[13] P. Welinder, S. Branson, S. Belongie, and P. Perona. The multidimensional wisdom of crowds. In *NIPS*, 2010.

[14] P. Welinder and P. Perona. Online crowdsourcing: Rating annotators and obtaining cost-effective labels. In *Computer Vision and Pattern Recognition Workshop*, 2010.

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

[16] H. Yu, M. Kaminsky, P. B. Gibbons, and A. D. Flaxman. Sybilguard: Defending against sybil attacks via social networks. *IEEE/ACM Trans. Netw.*, 16(3):576–589, 2008.

A Linear-algebraic Inequalities

Lemma A.1. *Let A and B be symmetric, positive semidefinite matrices such that $\|A - B\|_2 \leq \delta$. Let v_1 and w_1 be the unit eigenvectors corresponding to the top eigenvalues of A and B respectively. Let λ_1 and λ_2 be the top two eigenvalues of B. Then*

$$(v_1 \cdot w_1)^2 \geq 1 - \frac{\lambda_2 + 3\delta}{\lambda_1}.$$

Proof. Let μ_1 and μ_2 be the top two eigenvalues of A. Lemma A.2 below implies that $|\mu_1 - \lambda_1| \leq \delta$ and $|\mu_2 - \lambda_2| \leq \delta$. Let $w_1 = \alpha v_1 + x$ be a decomposition of w_1 along v_1 and its orthogonal complement, so that $\alpha = v_1 \cdot w_1$ and $\|x\| = \sqrt{1 - \alpha^2} \leq 1$. Since $\|A - B\| \leq \delta$, we have

$$w_1^\top A w_1 \geq w_1^\top B w_1 - \delta = \lambda_1 - \delta.$$

On the other hand we have

$$w_1^\top A w_1 = \alpha^2 v_1^\top A v_1 + x^\top A x \leq \alpha^2 \mu_1 + \mu_2 \|x\|^2 \leq \alpha^2(\lambda_1 + \delta) + (\lambda_2 + \delta).$$

Putting the two inequalities together, we get that

$$\alpha^2 \geq 1 - \frac{\lambda_2 + 3\delta}{\lambda_1 + \delta} \geq 1 - \frac{\lambda_2 + 3\delta}{\lambda_1}.$$

as required. □

Lemma A.2. *Let A and B be $T \times T$ symmetric matrices such that $\|A - B\|_2 \leq \delta$. Let μ_i and λ_i denote the i^{th} largest eigenvalues of A and B respectively, for $i = 1, \ldots, T$. Then, for all i, we have $|\mu_i - \lambda_i| \leq \delta$.*

Proof. Consider the subspace W_i spanned by the top i eigenvectors of A and the subspace V_{T-i+1} spanned by the bottom $T - i + 1$ eigenvectors of B. Since the sum of the dimensions of these two subspaces is $T + 1$, their intersection has a non-zero unit vector v. By the Courant-Fischer min-max theorem, we have

$$v^\top A^\top v \geq \mu_i, \text{ and } v^\top B v \leq \bar{\lambda}_i,$$

since $v \in W_i$ and $v \in \bar{V}_{T-i+1}$ respectively. Since $\|A - B\|_2 \leq \delta$, we conclude that $|\mu_i - \lambda_i| \leq \delta$. □

B Omitted proofs

We restate and prove Theorem 3.2 now:

Theorem B.1. *There is a constant c such that if $T > \frac{3}{p_1 \gamma^2} \log(4/\eta)$ and $\frac{n}{\log(n)} > \frac{88 p_1}{c \mathcal{S}^2}$, then for any $\eta \in (0, 1)$, with probability at least $1 - \eta$, we have*

$$\frac{1}{T} |\{t : \ q_t \neq q_t'\}| \leq 6 \sqrt{\frac{\log(n) \log(4/\eta)}{c \bar{\kappa}_p \sum_i p_i}} + \frac{2}{\bar{\kappa}_p T}. \tag{9}$$

The proof relies on the the following analogue of Lemma 3.3:

Lemma B.1. *If $T > \frac{3}{p_1 \gamma^2} \log(4/\eta)$ and $\frac{n}{\log(n)} > \frac{88 p_1}{c \bar{\kappa}^2}$, then with probability at least $1 - \eta/2$, we have $\hat{q} \cdot (\sigma v) \geq 0$.*

Proof. The proof is on the same lines as the proof of Lemma 3.3. The only difference is that we use Bernstein's inequality instead of Hoeffding's inequality to get a tighter bound: for T independent random variables $X_1, X_2, \ldots, X_T \in \{-1, 1\}$, for any $\delta > 0$, we have

$$\Pr\left[\sum_{t=1}^{T} X_t - \mathbb{E}\left[\sum_{t=1}^{T} X_t\right] \leq \delta\right] \leq \exp\left(\frac{-\delta^2/2}{\sum_{t=1}^{T} Var(X_t) + \delta/3}\right).$$

In our case, setting $X_t = u_{1t}q_t$, we have $\mathbb{E}[\sum_{t=1}^{T} X_t] = \mathbb{E}[u_1 \cdot q] = 2p_1\gamma T$. Further, $Var(X_t) \leq \mathbb{E}[X_t^2] = p_1$. Setting $\delta = p_1\gamma T$, we get that

$$\Pr[u_1 \cdot q < p_1\gamma T] < \exp(-p_1\gamma^2 T/3).$$

The rest of the analysis is as before. □

We restate and prove Lemma 3.4:

Lemma B.2. *Let $\epsilon' = \Theta(\epsilon^c)$ for some constant c. Let x be a randomly chosen vector in $\{-1, 1\}^n$. Let $y = (UU^\top)^k x$, Then for a large enough constant k, with probability at least $1/8$, we have $(v \cdot \hat{y}) \geq 1 - \epsilon'$.*

Proof. Let the eigenvalues of $UU^\top$, in decreasing order, be $\mu_1, \mu_2, \ldots, \mu_T$, and let $v_1 = v, v_2, \ldots, v_T$ be the corresponding unit eigenvectors. Express x in terms of the eigenvectors as $x = \sum_t \alpha_t v_t$. Then $y = (UU^\top)x = \sum_t \alpha_t \mu_t^k v_t$, and so

$$(v \cdot \hat{y})^2 = \frac{\alpha_1^2 \mu_1^k}{\sum_t \alpha_t^2 \mu_t^k} = \frac{1}{1 + \sum_{t>1} \frac{\alpha_t^2}{\alpha_1^2}(\frac{\mu_t}{\mu_1})^{2k}} \geq \frac{1}{1 + \frac{T}{\alpha_1^2}(\frac{\mu_2}{\mu_1})^{2k}}.$$

Here we use the fact that $\sum_t \alpha_t^2 = \|x\|^2 = T$. Now, with probability at least $1/8$ [12], we have $\alpha_1^2 \geq 1/4$. Furthermore, by (3) we have $\|UU^\top - \mathbb{E}[UU^\top\| \leq \delta$. Applying Lemma A.2 with $A = UU^\top$ and $B = \mathbb{E}[UU^\top]$, we get that $\mu_2 \leq \lambda_2 + \delta$, and $\mu_1 \geq \lambda_1 - \delta$, where recall that λ_1 and λ_2 are the top eigenvalues of $\mathbb{E}[UU^\top]$. This implies that

$$\frac{\mu_2}{\mu_1} \leq \frac{\lambda_2 + \delta}{\lambda_1 - \delta} = O(\epsilon).$$

Hence, with probability at least $1/8$, we have

$$(v \cdot \hat{y})^2 \geq \frac{1}{1 + 4T \cdot O(\epsilon)^k} \geq 1 - \epsilon'$$

when $k = \Theta(\frac{\log(T/\epsilon')}{\log(1/\epsilon)}) = \Theta(1)$, since $\epsilon = O(\sqrt{1/T})$. □

We restate and prove Lemma 4.2 now:

Lemma B.3. *If $\alpha \leq 0.05$, the following bound holds:*

$$\left(\psi_i \cdot \sqrt{\frac{1 - \psi_i''}{\psi_i''}} + (1 - \psi_i) \cdot \sqrt{\frac{\psi''}{1 - \psi_i''}}\right) \leq \exp(-0.5(2\psi_i - 1)^2 + 12\alpha).$$

Proof. Without loss of generality we may assume that $\psi'' \leq \psi$; the other case is handled analogously with $1 - \psi$ and $1 - \psi''$ playing the roles of ψ and ψ'' respectively. Next, note that

$$|(2\psi_i - 1)^2 - (2\psi_i'' - 1)^2| = |4(\psi_i - \psi_i'')(\psi_i + \psi_i'' - 1)| \leq 4\alpha. \tag{10}$$

We prove the bound of the lemma via two cases on ψ_i:
Case 1: $\psi_i \geq 0.15$. Since $\psi_i'' \geq \psi_i - \alpha$, we can bound

$$\psi_i \leq (1 + \tfrac{\alpha}{\psi_i''})\psi_i'' \leq (1 + \tfrac{\alpha}{\psi_i - \alpha})\psi_i'' \leq (1 + 10\alpha)\psi_i''$$

since $\psi_i - \alpha \geq 0.1$ because $\alpha \leq 0.05$. So

$$\begin{aligned}\psi_i \cdot \sqrt{\frac{1-\psi_i''}{\psi_i''}} &\leq (1+48\alpha) \cdot \sqrt{\psi''(1-\psi'')} \\ &\leq 0.5\exp(-(2\psi_i''-1)^2 + 10\alpha) \\ &\leq 0.5\exp(-(2\psi_i-1)^2 + 12\alpha). \end{aligned} \tag{11}$$

The second inequality above uses the fact that

$$\sqrt{x(1-x)} \leq 0.5\exp(-0.5(2x-1)^2)$$

for all $x \in [0,1]$, and the third inequality uses (10). Further, since $\psi_i'' \leq \psi_i$ we have

$$\begin{aligned}(1-\psi_i) \cdot \sqrt{\frac{\psi''}{1-\psi_i''}} &\leq (1-\psi_i) \cdot \sqrt{\frac{\psi}{1-\psi_i}} \\ &= \sqrt{\psi(1-\psi)} \\ &\leq 0.5\exp(-0.5(2\psi_i-1)^2), \end{aligned} \tag{12}$$

using the inequality $\sqrt{x(1-x)} \leq 0.5\exp(-0.5(2x-1)^2)$ again. Putting (11) and (12) together, we get the desired bound in the statement of the lemma.

Case 2: $\psi < 0.15$**.** Let $r = (\frac{\psi_i}{1-\psi_i})/(\frac{\psi_i''}{1-\psi_i''}) = \frac{\psi(1-\psi_i'')}{\psi_i''(1-\psi_i)}$. Since $\psi_i'' \leq \psi_i$, we have $r \geq 1$.

We now wish to upper bound r. The ratio $\psi_i/\psi_i'' \leq 2$, since $\psi_i'' \geq \alpha$ and $\psi_i \leq \psi_i'' + \alpha$. The ratio $(1-\psi_i'')/(1-\psi_i) \leq 1 + \frac{\alpha}{0.85}$ since $1-\psi_i \geq 0.85$ and $1-\psi_i'' \leq 1-\psi_i+\alpha$. Overall, we get that $r \leq 2(1+\frac{\alpha}{0.85}) \leq 2.12$ since $\alpha \leq 0.05$.

We now have

$$\begin{aligned}\psi_i \cdot \sqrt{\frac{1-\psi_i''}{\psi''}} + (1-\psi_i) \cdot \sqrt{\frac{\psi''}{1-\psi_i''}} &= (1/\sqrt{r} + \sqrt{r})\sqrt{\psi(1-\psi)} \\ &\leq 2.15\sqrt{\psi(1-\psi)} \\ &\leq \exp(-0.5(2\psi_i-1)^2). \end{aligned}$$

Here, the second inequality uses the fact that for $r \in [1, 2.12]$, we have

$$(1/\sqrt{r} + \sqrt{r}) \leq (1/\sqrt{2.12} + \sqrt{2.12}) \leq 2.15.$$

The third inequality is based on the bound

$$2.15\sqrt{x(1-x)} \leq \exp(-0.5(2x-1)^2)$$

which is valid when $x < 0.15$. □