Aggregating Crowdsourced Binary Ratings

Nilesh Dalvi
Facebook, Inc.
Menlo Park, CA
nileshdalvi@gmail.com

Anirban Dasgupta
Yahoo! Labs
Sunnyvale, CA
anirban.dasgupta@gmail.com

Ravi Kumar
Google
Mountain View, CA
ravi.k53@gmail.com

Vibhor Rastogi
Google
Mountain View, CA
vibhor.rastogi@gmail.com

ABSTRACT

In this paper we analyze a crowdsourcing system consisting of a set of users and a set of binary choice questions. Each user has an unknown, fixed, reliability that determines the user's error rate in answering questions. The problem is to determine the truth values of the questions solely based on the user answers. Although this problem has been studied extensively, theoretical error bounds have been shown only for restricted settings: when the graph between users and questions is either random or complete. In this paper we consider a general setting of the problem where the user–question graph can be arbitrary. We obtain bounds on the error rate of our algorithm and show it is governed by the expansion of the graph. We demonstrate, using several synthetic and real datasets, that our algorithm outperforms the state of the art.

Categories and Subject Descriptors

H.4.m [**Information Systems**]: Miscellaneous

Keywords

Crowdsourcing, mechanical turk, spectral methods

1. INTRODUCTION

Ever since Amazon launched its Mechanical Turk in 2005, crowdsourcing and human computing have become part and parcel of the World-Wide Web experience (`en.wikipedia.org/wiki/Crowdsourcing`). The topic frequently hits popular media, ranging from plaudits[1] to all-round skepticism[2]. Crowdsourcing has also attracted the attention of the research community at large, as evinced by the number of workshops and tutorials in many recent conferences dedicated to this topic: WWW[3], WSDM[4], SIGIR[5], CHI[6], KDD/AAAI[7].

As its name suggests, crowdsourcing taps into the wisdom of crowds. In its most basic version, it involves posing a presumably hard question to a set of users and aggregating their individual responses in order to deduce the answer to the question. This simple paradigm is useful in two scenarios where human labeling offers some version of the ground truth. First, it can be used to generate large quantities of labeled examples for algorithms that are based on machine learning. Second, it can be used for large-scale human evaluation and comparison of different algorithms for a problem.

Even this simplest version of crowdsourcing already poses an interesting research challenge: how to aggregate the responses of the users in order to obtain the true answer to each question? Meticulous users can be more accurate than the others in answering the questions, whereas unreliable/lazy (or spammy) users can provide random (or even adversarial) answers. To further complicate the problem, in many such systems, the reliability of a user may not be known a priori; indeed, a large fraction of the users may even be new recruits. These issues entail a holistic approach to the problem: rather than aggregate the answers for each question in isolation, it becomes necessary to look at the global matrix of user provided answers to all the questions in order to simultaneously elicit both the user reliabilities and the true answers.

There have been several approaches [5, 10, 19, 2, 14, 3, 15, 11] to formalizing this problem. These approaches posit a set of items with *binary qualities*, and a set of users indicating the qualities of items. Not all users necessarily rate all items. A bipartite graph G between items and users captures the set of items rated by each user. Typically, a simple model is assumed for users: each user is associated with a reliability measure, which is used to independently "corrupt" her perception of the true quality of the item. Given a set of user ratings, the problem is to collectively determine the reliability of each user and the true quality of each item. These approaches fall into two broad categories: machine-learning based and linear-algebraic based. The machine-learning approaches are based on variants of EM, which work on any graph G, but offer no guarantees as to how well they perform (see Section 2).

Algebraic approaches, on the other hand, can provide theoretical guarantees on the error in estimating item qualities, but so far have been limited to either complete assignment graphs (when each user rates all items) or to random graphs (when the assignment of users to items is random). One of the first algebraic approaches was proposed by Ghosh et al. [5], who present an algorithm with the following guarantee: for a random user–item assignment graph with n users, where in expectation each user rates D items and each item receives Δ ratings, the fraction of incorrectly estimated items bounded by $O(\sqrt{\frac{n}{D^3}})$. This bound is vacuous for sparse graphs where each user rates $o(n^{1/3})$ items. Karger et al. [10] show that

[1] `sfgate.com/business/prweb/article/Crowdsourced-mobile-fraud-intervention-solution-4009930.php`

[2] `www.technologyreview.com/view/416966/how-mechanical-turk-is-broken/`

[3] `crowdsearch.como.polimi.it/`

[4] `ir.ischool.utexas.edu/csdm2011/`

[5] `ir.ischool.utexas.edu/cse2010/`

[6] `crowdresearch.org/chi2011-workshop/`

[7] `www.humancomputation.com`

WWW 2013, May 13–17, 2013, Rio de Janeiro, Brazil.
ACM 978-1-4503-2035-1/13/05.

for random graphs, in the limit when number of items is going to infinity, the error in item qualities can be asymptotically bounded by $e^{-O(\Delta)}$, where Δ is again the expected number of users rating an item. Thus their bound is stronger than [5] and holds for sparse random graphs as well, but only asymptotically.

Our work is motivated by the fact that the user–item graph G is in practice *neither random nor regular*. Often, users determine both the number as well as the set of items they want to rate. The former is a function of their motivation level while the latter is determined by their expertise and familiarity with the items. Under such circumstances, it is not obvious how the techniques developed in [5] and [10] generalize—e.g., [5] depends crucially on the fact that the "expected" item–item agreement matrix is low-rank and hence recoverable under random perturbations, which the assumed generative mechanism posits as the model for user mistakes. Similarly, the performance of [10] depends crucially on whether belief propagation converges in arbitrary graphs. Thus it remains an open question to develop a strategy for aggregating user ratings when we do not have too much control in deciding which users choose what set of items to rate—whether there are characteristics of the user–item rating graph that make it amenable to good aggregation.

Main results. Our main contribution is an eigenvector-based technique to estimate both the user reliabilities and the item qualities that works for *arbitrary* user–item assignment graphs G. We bound the error rate as a function of the expansion gap, i.e., the gap between the first and second eigenvalues of the graph G^tG. The essence of our technique is to look at the user–user agreement matrices—measuring agreement between pairs of users—that are normalized by the number of items they decided to co-rate, and to then extract its topmost eigenvector. A key element of our approach is to show a concentration result for structured random matrices, using the matrix version of McDiarmid's inequality. We then present two algorithms that are based on matrix completion; for each of the algorithms, we prove that the estimated user reliabilities are close to the truth if the graph G^tG has some expansion properties. If the assignment graph is random, our estimate for user reliabilities translates into an approximation for the item qualities as well.

In particular, for a (D, Δ)-regular graph with a large eigengap, our bounds translate into a user reliability estimation error of $\tilde{O}(\frac{1}{\sqrt{D}} + \frac{1}{\Delta})$. On the other hand, even if we knew the true answer to each of the D questions that a user responds to, the estimated user reliability would still have a variance of $1/D$, resulting in an estimation error of $\Omega(1/D)$ in the user reliabilities. Our error bound of $O(D^{-1/2})$ is not too far off from this lower bound. For (D, Δ)-random assignment graph our algorithm makes mistakes on only $e^{-O(\Delta)}$ fraction of the items. Our bound generalizes both the results of [5] and [10], since this result holds for sparse graphs (unlike [5]) without requiring the asymptotic argument of the number of items going to infinity (unlike [10]).

Finally we also demonstrate our algorithms on real world datasets and show how they improve upon the state of the art in terms of accuracy of estimates in both item qualities and user reliabilities.

2. RELATED WORK

Crowdsourcing, using the global marketplace to perform microtasks in a scalable way, is a topic that has generated much excitement [8, 12]—labeling and rating items consists of a large fraction of such tasks. A key problem in here is to decide how to aggregate the labels from multiple labelers of varying reliabilities such that the effect of the underlying noise is mitigated. The extensive empirical work by Sheng et al. [15] shows that getting more noisy labels per item and then aggregating them is more accurate than getting more expensive, and hence allegedly more "accurate", labels; their work uses only majority voting to aggregate labels from multiple users, and is primarily concerned with identifying the items that will benefit from more labels. Dekel et al. [4] show that such aggregation can be improved if the bad raters are pruned.

A more general analysis of the user reliabilities was done by Dawid et al. [3], who are the first to model the obfuscation of labels by judges, and use the EM algorithm in order to derive the true labels. Unfortunately, the EM technique suffers from lack of theoretical guarantees and has issues regarding convergence and initialization. Since then, there has been a host of followup work modifying this approach using a Bayesian technique [2, 11], studying it in the context of learning a specific classifier [14], and modifying it by finding out spammers, i.e., labelers deliberately giving incorrect responses [13]. Other related results in applying machine learning techniques to cleaning user labels include [19, 16, 14, 18].

Much of the above work does not come with theoretical guarantees on the inferred user reliabilities or the item labels. Both Ghosh et al. [5] and Karger et al. [10] study this problem independently in the same generative setting, where each user rates a random set of items, and has an inherent probability of identifying the correct label, or flipping it. Our model is essentially a generalization of their setting to arbitrary user–item assignment graphs. Ghosh et al. [5] present a spectral algorithm that provably learns the true item qualities, with bounded error. However, as pointed out, these bounds are useful only when each user performs a large number of ratings. Karger et al. [10] uses belief propagation[8] to derive both a set of user reliabilities and an estimate for item qualities for a sparse random graph. Their convergence analysis uses techniques from density evolution and hence critically depends on the fact the graph is both sparse and random. Liu et al. [11] extend the BP algorithm of [10] via a Bayesian approach by choosing a suitable prior for item qualities and user reliabilities, and uses clever techniques to make the message passing more efficient.

An orthogonal question to ours, and one that has received much attention, is how to design incentives such that each user performs to the best of his abilities and provides truthful ratings [6, 9].

3. PROBLEM DESCRIPTION

Let m be the number of *items* and n the number of *users*. Let $q_i \in \{-1, 1\}$ denote the *quality* of the ith item. Let q denote the column vector of length m with q_i as the ith entry. Each user rates a subset of items. Let $G \in \{0, 1\}^{m \times n}$ denote the item–user *assignment* matrix, i.e., $G_{ij} = 1$ if item i is rated by user j.

3.1 Rating generation model

The ratings given by n users on m items is represented by a stochastic matrix U generated by the following random process (similar to [5]). Each user j is associated with a probability $p_j \in [0, 1]$ that captures how correct is her rating. Independently, for each item i she rates (as dictated by G), she tosses a coin with bias p_j: with probability p_j, she rates item i (correctly) as q_i and with probability $1 - p_j$, she rates item i (incorrectly) as $-q_i$. Thus, the random matrix $U \in \{-1, 0, 1\}^{m \times n}$ can be described as

$$U_{ij} = \begin{cases} q_i & \text{if } G_{ij} = 1, \text{ w.p. } p_j, \\ -q_i & \text{if } G_{ij} = 1, \text{ w.p. } 1 - p_j, \\ 0 & \text{if } G_{ij} = 0. \end{cases} \quad (1)$$

[8] Belief Propagation (BP) operates on the user–item bipartite graph, and like any standard BP algorithm, excludes the message from the node when computing the outgoing message to that node—if this message is included, then the algorithm reduces to that of [5].

We call this random process as *rating generation*. Let $w_j = 2p_j - 1$; we call w_j the *reliability* of user j. Thus, the user reliabilities are in the range $[-1, 1]$, where a reliability of 1 indicates a user who always answers correctly, a reliability of -1 indicates one who always answers incorrectly, and a reliability of 0 indicates one who answers uniformly at random. Let $w \in \Re^n$ denote the vector of user reliabilities.

3.2 Problem definition

The algorithm is given as input a realization of the stochastic rating matrix U, assumed to be generated from the set of latent parameters q and w, which are unknown. The aim is to estimate both the user reliabilities and the item qualities simultaneously, i.e., an estimate $\hat{w} \in [-1, 1]^n$ for the user reliabilities and an estimate $\hat{q} \in \{-1, 1\}^m$ for the item qualities. The performance of the algorithm is measured by the distance to the underlying reliability vector and the quality vector. The *errors* for the estimates $\hat{w}$ and $\hat{q}$ provided by the algorithm will thus be defined by the following quantities: $\mathrm{error}(\hat{w}) = \frac{1}{n}\mathbb{E}(||\hat{w} - w||_2^2)$ and $\mathrm{error}(\hat{q}) = \frac{1}{m}\mathbb{E}(||\hat{q} - q||_2^2) = \frac{4}{m}\mathbf{1}[\hat{q}_i \neq q_i]$.

4. TECHNIQUES

We review some definitions from linear algebra before presenting our algorithms.

4.1 Background

Throughout the paper, we represent (column) vectors using lowercase letters $(a, b, w, \ldots)$ and matrices by uppercase letters $(M, N, \ldots)$. Let $x \cdot y$ denote the innerproduct of x and y and let x^t denote the transpose of x. For a matrix M, the spectral and Frobenius norms are denoted by $\|M\| = \|M\|_2 = \max_{\|x\|_2=1} \|Mx\|_2$ and $\|M\|_F = (\sum_{ij} M_{ij}^2)^{1/2}$ respectively. For two matrices M and N of matching dimensions, we define the following *Hadamard products*:

$$\begin{aligned}(M \otimes N)_{ij} &= M_{ij}N_{ij};\\ (M \oslash N)_{ij} &= \begin{cases} M_{ij}/N_{ij} & \text{if } N_{ij} \neq 0,\\ 0 & \text{otherwise.}\end{cases}\end{aligned}$$

For any matrix $M \in \Re^{m\times n}$, we define the *indicator* matrix $\mathbb{I}(M) \in \{0, 1\}^{m\times n}$ such that $\mathbb{I}(M)_{ij} = \mathbf{1}[M_{ij} \neq 0]$, i.e., $\mathbb{I}(M)_{ij}$ is 1 if and only if the corresponding entry of M is nonzero. We will also denote the (scaled) top eigenvector of a matrix M as $v_1(M) = \arg\min_x \|M - xx^t\|_2, x^{(1)} \geq 0$.

We use the convention that indices i always denote items and indices $j, k, \ldots$ denote users. Let δ_i denote the number of ratings that item i gets, i.e., the number of non-zero entries in row i in G. Similarly, let d_j denote the number of ratings that user j supplies. Define $D = \max_j d_j$ and $\Delta = \max_i \delta_i$.

4.2 Algorithms

Recall that we are only presented with a realization of the ratings matrix U (and hence its indicator, the assignment matrix G as well) and we need to estimate both the item qualities and the user reliabilities. Before describing the algorithms, we present the intuition behind them.

The main idea is to work with the two user–user matrices U^tU and G^tG. The entry $(G^tG)_{jk}$ is the number of common items rated by users j and k. The entry $(U^tU)_{jk}$ is the difference between the number of agreements and disagreements of the users j and k. Let E denote the matrix which contains itemwise expected values of the random matrix U, i.e., $E_{ij} = \mathbb{E}[U_{ij}] = (p_jq_i + (1 - p_j)(-q_i))G_{ij} = q_iG_{ij}w_j$.

```
Input: $U \in \{-1, 0, 1\}^{m\times n}$, $G \in \{0, 1\}^{m\times n}$.
Output: $\tilde{w} \in \Re^n$, $\hat{q} \in \Re^m$.
$\hat{w} = v_1(U^tU) \oslash v_1(G^tG)$
Define $\tilde{w}_j = \mathrm{sgn}(\tilde{w_j})\max(|\hat{w}_j|, 1)$
Define $\hat{q}_i = \mathrm{sgn}(\sum_j U_{ij}\tilde{w}_j)$.
Output $\hat{w}$, $\hat{q}$.
```

Algorithm 1: Ratio of eigenvectors.

It is easy to see that

$$\begin{aligned}E^tE &= (G^tG) \otimes (ww^t),\\ E^tE \oslash G^tG &= \mathbb{I}(G^tG) \otimes ww^t. \end{aligned} \qquad (2)$$

Suppose we knew the expected matrix E. We could then estimate the user reliabilities by solving the following problem

$$\begin{aligned}F(A, B) = \arg\min_w \|A - B \otimes (ww^t)\|_F,\\ \text{s.t.}\ \forall_j, \quad w_j^2 \leq 1. \end{aligned} \qquad (3)$$

with (A, B) as either (E^tE, G^tG) or $(E^tE \oslash G^tG, \mathbb{I}(G^tG))$. It is easy to see why this approach works if the graph G is complete: the expected matrix $E = qw^t$ and solving (3) would give us back the user reliabilities w exactly. This approach, however, has a few problems when the graph G is arbitrary. First, the above program is computationally intractable for arbitrary G (e.g., [7]). But more importantly, we show that for arbitrary assignment graphs the matrix E^tE might not be informative, as shown in the following example.

Suppose there were two disjoint user groups A and B, and a user $x \notin A \cup B$. All users in A have reliability 1, those in B have reliability -1, and user x has reliability 0. The items have two disjoint groups S and T of size $m/2$. All users in A rate all items in S, all users in B rate all in T, and user x rates all items in $S \cup T$. It is clear that by looking only at the matrix E^tE, it is not possible to distinguish the highly reliable users from the non-reliable ones. It is easy to extend this construction to $k + 2$ user partitions such that we cannot distinguish the high and low proficiency users even if we are explicitly given, in addition to E^tE, the names of k users who answer all the questions. Thus, we want to characterize the class of graphs G that allows us to recover w with small errors [9].

One of our main contributions is to identify the expansion of the graph G as a sufficient property that enables us to estimate w both efficiently and with low errors— the resulting algorithms are presented in Algorithm 1 and 2. Since the matrix E is not observable, we instead work with the matrix U. Algorithm 1 is inspired by the observation that when G^tG has rank one, (3) has an exact solution $\hat{w}$ where $\hat{w} \otimes v_1(G^tG) = v_1(E^tE)$ and hence $\hat{w} = v_1(E^tE) \oslash v_1(G^tG)$. We will show that when the graph G has sufficiently high expansion, this solution, even when using U^tU in place of E^tE, is a reasonable approximation. Algorithm 2 is inspired by (2) and uses the same intuition that (3) is approximable when $\mathbb{I}(G^tG)$ is close to a rank one matrix. Hence, in this case, we first compute the rank one approximation $v_1(\mathbb{I}(G^tG))$ and then use it to compute the final estimate $\hat{w}$.

We next show an error bound on the estimate $\hat{w}$ for user reliability obtained from Algorithm 1. (Similar bounds can be shown for Algorithm 2, which we defer to the full version.) Our error bound holds for arbitrary graphs having expansion properties. However,

[9] Previous approaches have looked at the matrix UU^t (as a proxy for EE^t) [5]; by augmenting the above construction it is possible to show that such approaches will also incur a constant fraction error for arbitrary assignment graphs.

```
Input: $U \in \{-1, 0, 1\}^{m \times n}$, $G \in \{0, 1\}^{m \times n}$.
Output: $\tilde{w} \in \Re^n$, $\hat{q} \in \Re^m$.
Define $\hat{w} = v_1(U^tU \oslash G^tG \oslash v_1(\mathbb{I}(G^tG))$
Define $\tilde{w}_j = \mathrm{sgn}(\hat{w}_j) \max(|\hat{w}_j|, 1)$
Define $\hat{q}_i = \mathrm{sgn}(\sum_j U_{ij}\tilde{w}_j)$.
Output $\hat{w}$, $\hat{q}$.
```

Algorithm 2: Eigenvectors of ratio.

in order to illustrate our bounds, we state the results for (D, Δ)-regular graphs. The more general result is stated and proved in Section 5 (See Theorem 5.12). Let $\bar{w} = \sqrt{\frac{\sum_i w_i^2}{n}}$ denote the average reliability of users.

THEOREM 4.1 (USER ERROR BOUND). *Let $\epsilon, \delta < 1$ be a fixed positive constants. If G is a (D, Δ)-regular graph such that $\Delta > \frac{1}{8\epsilon\bar{w}^2}$, $D > \frac{256\log(n/\delta)}{\epsilon^2\bar{w}^2}$ and the second eigenvalue of G^tG, denoted by μ_2, satisfies the condition $\mu_2 < \frac{\epsilon\bar{w}^2 D\Delta}{16}$, then with probability $1 - \delta$, Algorithm 1 returns an estimate $\hat{w}$, such that*

$$\mathrm{error}(\hat{w}) = \tilde{O}\left(\frac{1}{\Delta} + \frac{1}{\sqrt{D}}\right).$$

When the item–user assignment graph is random, this error bound translates into a bound for error in item estimates. The question of whether such a bound holds for fixed graphs, under some assumptions, remains open.

THEOREM 4.2. *Let G be a random (D, Δ)-regular graph. With high probability, Algorithm 1 returns estimates $\hat{q}$, such that*

$$\mathrm{error}(\hat{q}) \leq \exp\left(-O\left(\Delta\left(\bar{w}^2 - \frac{1}{\Delta} - \frac{1}{\sqrt{D}}\right)^2\right)\right).$$

When the average reliability $\bar{w}$ is some constant bounded away from 0 (i.e., users are good on average), then $\mathrm{error}(\hat{q})$ scales as $\exp(-O(\Delta))$. This matches the bound in [10]. However, the bound in [10] requires that the limit of number of items goes to infinity, an assumption we no longer require.

4.3 Alternate projections

So far we have considered the case when G has a large expansion gap, i.e., when the second eigenvalue is much smaller than the first. We propose a heuristic, without any theoretical guarantees, that improves the performance of both Algorithms 1 and 2 for low expansion graphs. This heuristic is based on the standard alternating projections technique [1] for solving the weighted low rank approximation problem.

Recall that we are trying to find a user reliability vector w as a solution to the problem $F(A, B) = \arg\min_w \|A - B \otimes (ww^t)\|_F$. When B and $\mathbb{I}(B)$ satisfy expansion properties, Algorithm 1 and 2 both give good approximations to this problem. Consider a slight generalization of this problem that instead finds two vectors u and v to minimize $\arg\min_{u,v} \|A - B \otimes (uv^t)\|_F$. When one of the vectors, say u, is known, the other can be computed by solving a simple least squares problem. Thus, this gives an EM-style alternating projections algorithm to iteratively compute u and v. On convergence, we are guaranteed to achieve a local optimum, which for symmetric matrices A and B implies that $u = v$. This common converged value can thus be used instead of w.

One problem with this approach is that since the original problem is not convex, the convergence can happen at a local minima. Thus, both the rate of convergence and the quality of converged solution depends on the initialization for u and v. In practice we observed that when $u = v = \hat{w}$, where $\hat{w}$ is the estimate obtained by either Algorithm 1 or 2, then both rate of convergence and quality of converged solution is good. Intuitively, this is because Algorithm 1 and 2 already try to minimize the objective function (at least in the case of graphs with good expansions) and hence provide a very good seed for the alternating projections heuristic.

5. ANALYSIS

In this section we prove guarantees on the performance of our algorithms both in terms of the error incurred in estimating user reliabilities as well as for item qualities. The underlying intuition behind the proof is as follows. First we show that the response matrix U^tU is close to the expectation matrix E^tE. In order to prove this concentration bound, we need to use machinery aimed towards giving Chernoff-like tail bounds for sums of random matrices. We then use the expansion (and corresponding eigenvalue gap) of the user–user co-rating graph G^tG to show that the gap between the first and second eigenvalues of G^tG translates to a corresponding gap between the first and second eigenvalue of E^tE as well. Using this, we then characterize the top eigenvector of E^tE in terms of the top eigenvector of G^tG and the reliability vector w; the error in this characterization depends, among other quantities, on the ratio between the top two eigenvalues of the graph G^tG. This enables us to use the eigenvalues of G^tG and U^tU to create $\hat{w}$, an estimate of w. After creating an estimate $\hat{w}$ of the user reliabilities, we can then use it to create an estimate of the item qualities $\hat{q}$—the error in $\hat{q}$ will depend on the error in $\hat{w}$.

5.1 Matrix tail bounds

We start with a statement of the matrix McDiarmid inequality that we will use as a tool. The underlying intuition behind this concentration result from [17] is that a random matrix is close to its expectation in terms of the spectral norm, if it can be expressed as the output of a function having bounded sensitivity over its input variables. Note that $A \preceq B$ denotes the usual semidefinite ordering, i.e., $B - A$ is semidefinite.

THEOREM 5.1 (MATRIX BOUNDED DIFFERENCE [17]). *Let $\{Z_k\}_{k=1}^n$ be an independent family of random variables, and let H be a function that maps n variables to a self-adjoint matrix of dimension d. Consider a sequence $\{A_k\}$ of fixed self-adjoint matrices that satisfies*

$$(H(z_1, \ldots, z_k, \ldots, z_n) - H(z_1, \ldots, z'_k, \ldots, z_n))^2 \preceq {A_k}^2,$$

where z_i and z'_i range over all possible values of Z_i for each index i. Compute the variance parameter $\sigma = \|\sum_k {A_k}^2\|_2$. Denote the random vector $\mathbf{z} = (Z_1, \ldots, Z_n)$. Then, for all $t \geq 0$,

$$\Pr[\|H(\mathbf{z}) - \mathbb{E}[H(\mathbf{z})]\| > t] \leq d \cdot e^{-t^2/8\sigma^2}.$$

We will use Theorem 5.1 to show that the user–user agreement matrix U^tU is close to its expectation E^tE in the following sense. For a user j, denote $\rho_j = \sum_{i=1}^m G_{ij}\delta_i^2$, i.e., ρ_j is the sum of squared degrees of items that j responds to, and denote $\rho = \max_{j=1}^n \rho_j$. We first define the function $H(\cdot)$. Lemma 5.2 then characterizes the structure of the difference matrices when any of the random variables is perturbed. Using this structural characterization Lemma 5.3 shows that function $H(\cdot)$ satisfies the sensitivity conditions of Theorem 5.1, and Lemma 5.4 shows the final bound that we get using the sensitivity condition derived in Lemma 5.3.

We abuse notation and define the sequence of random variables

$$U = \{U_{11}, \ldots, U_{1n}, U_{21}, \ldots, U_{2n}, \ldots, U_{m1}, \ldots, U_{mn}\}.$$

The function $H(\cdot)$ is then defined as $H(U) = U^tU$, which is a self-adjoint matrix in $\Re^{n\times n}$. We also define the sequence of self-adjoint matrices $\{A_{ij} \in \Re^{n\times n}, i \in [m], j \in [n]\}$ where each A_{ij} is a diagonal matrix with kth diagonal entry as $\sqrt{8G_{ik}G_{ij}(\delta_i - 1)}$ for all $k \in [n]$. Lastly, we define column vectors e_j and r_{ij} of length n as following: e_j is the unit vector with 1 as the jth element, and $r_{ij}[k] = -2U_{ij}U_{ik}$ if $k \neq j$, and 0 otherwise.

The following Lemma shows the structure of the sensitivity matrices.

LEMMA 5.2. *For any response matrix U, denote $\Delta_{ij} = H(U) - H(U')$, where U' is the response matrix identical to U in all entries except with the (i,j)th entry switched, i.e., $U'_{ij} = -U_{ij}$ and $U'_{kl} = U_{kl}$ for $(k,l) \neq (i,j)$. Then $\Delta_{ij}^2 = r_{ij}r_{ij}^t + 4(\delta_i - 1)G_{ij}e_je_j^t$.*

PROOF. Recall that $H(U) = U^tU$ is an $n \times n$ matrix with the (j,j)th diagonal entry d_j, where d_j is the number of items rated by user j. Also $H(U)_{kl} = a_{kl} - b_{kl}$ where (b_{kl}) a_{kl} denotes the number of (dis-) agreements between users k and l in rating the items that they have in common.

Now since $\Delta_{ij} = H(U) - H(U')$, where U' differs from U only in the (i,j)th entry, Δ_{ij} is again an $n \times n$ matrix with all but the jth row and column as 0. To see why, consider (k,l)th entry of Δ_{ij} such that $k \neq j$ and $l \neq j$. Both users k and l have same responses in both U and U'. Thus the number of agreements and disagreements between k and l is same in U and U'. Hence the (k,l)th entry of Δ_{ij} is zero.

Since $H(U) = U^tU$ and $H(U') = U'^tU'$ are symmetric matrices, so is their difference Δ_{ij}. Thus, the jth row and column for Δ_{ij} are identical. We will show that the column is precisely the vector r_{ij} (and hence row is r_{ij}^t). Consider the kth element of this row. If user k has rated item i, and k and j agree according to U, then they will disagree according to U'. Similarly, if they disagree according to U, then they will agree according to U'. Thus, kth element of r_{ij}, which is the difference in agreements and disagreements of users k and j will change by either 2 or -2. These cases can be summarized succinctly as $-2U_{ij}U_{ik} = r_{ij}[k]$, by definition. Only exception is $r_{ij}[j]$, which is always 0, since no user disagrees with himself on the same item i. Thus the jth column of Δ_{ij} is precisely r_{ij}.

The fact that Δ_{ij} is the matrix with jth row and column equal to r_{ij} and rest elements as zero can be written as

$$\Delta_{ij} = r_{ij}e_j^t + e_jr_{ij}^t,$$

which yields that

$$\begin{aligned}
\Delta_{ij}^2 &= (r_{ij}e_j^t)(r_{ij}e_j^t) + (r_{ij}e_j^t)(e_jr_{ij}^t) \\
&\quad + (e_jr_{ij}^t)(r_{ij}e_j^t) + (e_jr_{ij}^t)(e_jr_{ij}^t) \\
&= r_{ij}(e_j^tr_{ij})e_j^t + r_{ij}(e_j^te_j)r_{ij}^t \\
&\quad + e_j(r_{ij}^tr_{ij})e_j^t + e_j(r_{ij}^te_j)r_{ij}^t \\
&= 0 + (1)r_{ij}r_{ij}^t + 0 + 4(\delta_i - 1)G_{ij}e_je_j^t.
\end{aligned}$$

Here, the last equation follows from using the following values of the four innerproducts highlighted in penultimate equation: $r_{ij}^te_j = e_j^tr_{ij}$ is 0 (since e_j has only jth entry as non-zero, which is zero in r_{ij}), $e_j^te_j$ is 1, and $r_{ij}^tr_{ij}$ is $4G_{ij}(\delta_i - 1)$ (since $\sum_k r_{ij}[k]^2 = \sum_{k\neq j}(-2U_{ik}U_{ij})^2 = \sum_{k\neq j} 4G_{ij}G_{ik} = 4(\delta_i - 1)G_{ij}$). The proof follows. □

Let $A_{ij} \in \Re^{n\times n}$ be defined as a diagonal matrix where the kth entry equals $\sqrt{8G_{ik}G_{ij}(\delta_i - 1)}$. Using the above lemma, we can show that Δ_{ij}^2 is bounded by the matrix A_{ij}.

LEMMA 5.3. $\Delta_{ij}^2 \preceq A_{ij}^2$.

PROOF. From Lemma 5.2, $\Delta_{ij}^2 = r_{ij}r_{ij}^t + 4(\delta_i - 1)G_{ij}e_je_j^t$. Now if we show that $r_{ij}r_{ij}^t \preceq A_{ij}^2/2$, then the proof of lemma is complete, since trivially, $4(\delta_i - 1)G_{ij}e_je_j^t \preceq A_{ij}^2/2$.

To show $r_{ij}r_{ij}^t \preceq A_{ij}^2/2$, consider the (k,l)th element, denoted by R_{kl}, of $r_{ij}r_{ij}^t$. If $k, l \neq j$, then we have

$$R_{kl} = (-2U_{ij}U_{ik})(-2U_{ij}U_{il}) = 4G_{ij}U_{ik}U_{il},$$

and hence

$$|R_{kl}| = 4G_{ij}|U_{ik}||U_{ij}| = 4G_{ij}G_{ik}G_{il}.$$

If either $k = j$ or $l = j$, then the (k,l)th element is 0. Hence for the kth row, the sum of the absolute values of (k,l)th entries is

$$\sum_l |R_{kl}| = \sum_l 4G_{ij}G_{ik}G_{il} = 4G_{ij}G_{ik}(\delta_i - 1),$$

since each user $l \neq j$ who rated item i contributes exactly 1 to the sum.

Thus the diagonal matrix with $4G_{ij}G_{ik}(\delta_i - 1)$ as the kth diagonal entry, diagonally dominates $r_{ij}r_{ij}^t$. Now $A_{ij}^2/2$ by definition is precisely such a diagonal matrix. Hence $r_{ij}r_{ij}^t \preceq A_{ij}^2/2$. □

The next statement shows that U^tU is close to the expectation matrix E^tE. Recall that $\rho = \max_j \sum_{i=1}^m G_{ij}\delta_i^2$.

LEMMA 5.4. *Suppose the matrix U is generated by the rating generation process described above. Then, for every $\delta \in (0,1)$,*

$$\Pr\left[\|U^tU - \mathbb{E}[U^tU]\|_2 \leq 8\sqrt{\rho \log(n/\delta)}\right] \geq 1 - \delta.$$

PROOF. Using the statement of Lemma 5.2, we get that the sensitivity of $H(\cdot)$ with respect to each variable U_{ij} is bounded by A_{ij}^2. Thus, from the statement of Theorem 5.1, the variance parameter σ is given by

$$\sigma^2 = \left\| \sum_{i=1}^m \sum_{j=1}^n A_{ij}^2 \right\|.$$

Since each A_{ij}^2 is diagonal, so is this sum. The kth diagonal entry of A_{ij}^2 is $8G_{ik}G_{ij}(\delta_i - 1)$ and hence the kth diagonal entry of the sum is given by

$$\begin{aligned}
\sum_{i=1}^m \sum_{j=1}^n 8G_{ik}G_{ij}(\delta_i - 1) &= \sum_{i=1}^m 8(\delta_i - 1)G_{ik} \sum_{j=1}^n G_{ij} \\
= \sum_{i=1}^m 8(\delta_i - 1)G_{ik}\delta_i &\leq 8\sum_{i=1}^m G_{ik}\delta_i^2 = 8\rho_k.
\end{aligned}$$

Hence the spectral norm, which is the largest diagonal entry for a diagonal matrix, is simply $8\max_k \rho_k = 8\rho$ and hence $\sigma^2 = 8\rho$. Using this value for σ, setting $d = n$, and $t^2 = 8\sigma^2 \log(n/\delta) = 64\rho\log(n/\delta)$ in Theorem 5.1 completes the proof. □

Finally, this implies the following result.

LEMMA 5.5. *For a matrix U generated by the random rating generation process, with probability $1-\delta$, and $E = \mathbb{E}[U]$, $\|U^tU - E^tE\| \leq 8\sqrt{\rho\log(n/\delta)} + D$, where D is the maximum number of ratings done by a person.*

PROOF. Assuming the result of Lemma 5.4 holds, we only need to bound the norm of $E^tE - \mathbb{E}[U^tU]$. This is a diagonal matrix, with the jth diagonal entry to be $d_j(1 - w_j^2)$. Hence, $\|E^tE - \mathbb{E}[U^tU]\| \leq \max_j d_j^2 = D$. □

5.2 Analysis of estimators

In this section we show that the estimators for user reliabilities and item qualities have a small error. For notational simplicity, we assume that the event in Lemma 5.4 holds, i.e., the matrix U^tU is close to its expectation.

5.2.1 Algorithm 1: Ratio of eigenvectors

We first show the proof for Algorithm 1, which takes the ratio of the top eigenvectors of U^tU and G^tG. The proof strategy is to first show that under a suitable set of assumptions for G, the matrix E^tE has a large gap between the first and second eigenvalues, and hence can be represented accurately using only the topmost eigenvector—this will ensure that the eigenvector-based Algorithm 1 has small error.

Let the first and second eigenvalues of G^tG be denoted by μ_1 and μ_2 respectively, and the top two eigenvalues of E^tE be denoted by λ_1 and λ_2. Let g denote the first eigenvector of G^tG, and e be that of E^tE. Let $g_{\min}$ denote the minimum entry of g; by Perron–Frobenius theorem, $g_{\min} > 0$. Recall $\bar{w}^2 = \frac{1}{n}\sum_j w_j^2$. Define $\kappa = \|U^tU - E^tE\|_2$ and $W \in n \times n$ to be the diagonal matrix with w_j for the jth diagonal entry.

LEMMA 5.6. $\lambda_1 \geq \mu_1\|Wg\|^2 - \mu_2$.

PROOF. Recall that $E^tE = (G^tG) \otimes (ww^t)$. Since μ_1 and g are the first eigenvalue and vector of G^tG, we have that $G^tG = \mu_1 gg^t + A$, where A is the matrix defined as the difference between G^tG and $\mu_1 gg^t$. Thus, $\|A\| = \mu_2$.

$$\begin{aligned} E^tE &= (G^tG) \otimes (ww^t) = (\mu_1 gg^t + A) \otimes (ww^t) \quad (4) \\ &= \mu_1 (Wg)(Wg)^t + A \otimes (ww^t). \end{aligned}$$

Hence we can write using the triangle inequality:

$$\begin{aligned} \|E^tE\| &\geq \|\mu_1 (Wg)(Wg)^t\| - \|A \otimes (ww^t)\| \\ &\geq \mu_1\|Wg\|^2 - \|A\| = \mu_1\|Wg\|^2 - \mu_2, \end{aligned}$$

where we use $\|A \otimes (ww^t)\| = \|WAW\| \leq \|W\|^2\|A\| \leq \|A\|$. This completes the proof. □

LEMMA 5.7. $(e^tWg)^2 \geq \|Wg\|^2 - \frac{2\mu_2}{\mu_1}$.

PROOF. From (4), we know that $E^tE = \mu_1(Wg)(Wg)^t + A \otimes (ww^t)$, where $\|A\| \leq \mu_2$. Also $eE^tEe^t = \lambda_1$ and

$$\begin{aligned} e^tE^tEe &= e^t(\mu_1(Wg)^t(Wg) + A \otimes (ww^t))e \\ &= \mu_1(e^tWg)^2 + e^t(A \otimes (ww^t))e \\ &\leq \mu_1(e^tWg)^2 + \mu_2, \end{aligned}$$

where the last inequality again follows from $\|A \otimes (ww^t)\| \leq \|A\|\|W\|^2 \leq \mu_2 \max_j w_j^2 \leq \mu_2$. Thus, we have

$$\lambda_1 = e^tE^tEe \leq \mu_1(e^tWg)^2 + \mu_2. \quad \square$$

LEMMA 5.8. $\lambda_2 \leq 3\mu_2$.

PROOF. Let x be the second eigenvector of E^tE. Then $xE^tEx^t = \lambda_2$. Also x is perpendicular to the largest eigenvector e of E^tE. So, we know $(e^tWg)^2 + (x^tWg)^2 \leq \|Wg\|^2$. From Lemma 5.7, we know $(e^tWg)^2 \geq \|Wg\|^2 - 2\mu_2/\mu_1$. Hence, $(x^tWg)^2 \leq 2\mu_2/\mu_1$. Thus, we can write

$$\begin{aligned} \lambda_2 &= x^tE^tEx = x^t\mu_1(Wg)(Wg)^t + A \otimes (ww^t) \\ &= \mu_1(x^tWg)^2 + x(A \otimes (ww^t))x \\ &\leq \mu_1 \cdot \frac{2\mu_2}{\mu_1} + \mu_2 = 3\mu_2. \quad \square \end{aligned}$$

LEMMA 5.9. *Let* $\kappa = \|U^tU - E^tE\|_2$ *and* $\tau = \|Wg\|$. *If* $\frac{\lambda_2+3\kappa}{\lambda_1} < 1$ *and* $\frac{2\mu_2}{\mu_1\tau^2} < 1$, *then*

$$\left\| u - \frac{Wg}{\tau} \right\| \leq \sqrt{2\frac{\lambda_2 + 3\kappa}{\lambda_1}} + \sqrt{\frac{4\mu_2}{\mu_1\tau^2}}.$$

PROOF. Since u and e are the top eigenvector of U^tU and E^tE respectively, and $\kappa = \|U^tU - E^tE\|$, by applying a standard matrix perturbation bound [5, Lemma 3.2],

$$(e \cdot u)^2 \geq 1 - \frac{\lambda_2 + 3\kappa}{\lambda_1}.$$

We write the bound derived in Lemma 5.7 as follows: $(e^t\frac{Wg}{\tau})^2 \geq 1 - \frac{2\mu_2}{\mu_1\tau^2}$. From the condition stated in the Lemma, since $2\mu_2 \leq \mu_1\tau^2$, and $\sqrt{1-x} \geq 1 - x$ for $0 < x < 1$, we have $e^t\frac{Wg}{\tau} \geq \sqrt{1 - \frac{2\mu_2}{\mu_1\tau^2}} \geq 1 - \frac{2\mu_2}{\mu_1\tau^2}$. Hence $\|e - \frac{Wg}{\tau}\|^2 = 2 - 2\frac{e^tWg}{\tau} \leq \frac{4\mu_2}{\mu_1\tau^2}$ Similarly, $e^tu \geq \sqrt{1 - \frac{\lambda_2+3\kappa}{\lambda_1}} \geq 1 - \frac{\lambda_2+3\kappa}{\lambda_1}$ and thus $\|e - u\|^2 \leq 2\frac{\lambda_2+3\kappa}{\lambda_1}$. The proof follows from the triangle inequality. □

LEMMA 5.10. *Denote* $\tau = \|Wg\|$ *and let* $\hat{w}$ *be the vector with the* i*th element* $\tau u_i/g_i$. *If* $\frac{\lambda_2+3\kappa}{\lambda_1} < 1$ *and* $\frac{2\mu_2}{\mu_1\tau^2} < 1$, *then*

$$\text{error}(\hat{w}) = \frac{\|\hat{w} - w\|^2}{n} \leq \frac{\tau^2}{ng_{\min}^2}\left(2\frac{\lambda_2 + 3\kappa}{\lambda_1} + \frac{4\mu_2}{\mu_1\tau^2}\right).$$

PROOF. From Lemma 5.9, we know that

$$\left\| u - \frac{Wg}{\tau} \right\| \leq \sqrt{2\frac{\lambda_2 + 3\kappa}{\lambda_1}} + \sqrt{\frac{4\mu_2}{\mu_1\tau^2}}.$$

Hence

$$\|\hat{w} - w\|^2 = \|(\tau u - Wg) \oslash g\|^2 \leq \frac{\tau^2\|u - Wg/\tau\|^2}{g_{\min}^2}.$$

Since $(\sqrt{x} + \sqrt{y})^2 \leq 2(x + y)$, we have

$$\|\hat{w} - w\|^2 \leq \frac{\tau^2}{g_{\min}^2}\left(2\frac{\lambda_2 + 3\kappa}{\lambda_1} + \frac{4\mu_2}{\mu_1\tau^2}\right),$$

and hence the proof. □

LEMMA 5.11. *Let* $\bar{w} = \sqrt{\frac{\sum_i w_i^2}{n}}$ *be the average reliability of users; let* $\tau = \|Wg\|$ *and* $r = g_{\max}/g_{\min}$. *Then,*

$$\tau \geq \bar{w}/r.$$

PROOF. This follows from considering the weighted graph corresponding to G^tG. Then

$$\sum_i w_i^2 g_i^2 \geq n\bar{w}^2 g_{\min}^2 \geq \bar{w}^2 n(g_{\max}^2/r^2) \geq \bar{w}^2/r^2,$$

which completes the proof. □

Combining the above lemmas, we get the final theorem about the error bounds.

THEOREM 5.12. *For a fixed assignment graph G and a rating matrix U that is generated by the random rating generating process, if the graph G satisfies*

$$\mu_2 < \frac{\mu_1\bar{w}^2}{4r} - 6\sqrt{\rho\log{(n/\delta)}} - D \quad (5)$$

then with probability $1 - \delta$, *Algorithm 1 returns estimates* $\hat{w}$, *such that*

$$\text{error}(\hat{w}) < \frac{10}{\mu_1 ng_{\min}^2}\left(\mu_2 + D + 5\sqrt{\rho\log{(n/\delta)}}\right).$$

PROOF. From Lemma 5.5, with probability $1 - \delta$,

$$\|U^tU - E^tE\|_2 \leq 8\sqrt{\rho \log{(n/\delta)}} + D.$$

Assume that the above event holds. Also, for Lemma 5.10, we need the following bounds:

$$\frac{\lambda_2 + 3\kappa}{\lambda_1} < 1, \quad \frac{2\mu_2}{\mu_1\tau^2} < 1. \tag{6}$$

Using the bounds on λ_1, λ_2 and κ, the above bounds are satisfied if

$$\begin{aligned} \mu_2 &< \frac{\mu_1\tau^2}{4} - 6\sqrt{\rho \log{(n/\delta)}} - D \\ &< \frac{\mu_1\bar{w}^2}{4r} - 6\sqrt{\rho \log{(n/\delta)}} - D. \end{aligned} \tag{7}$$

Conditioned on this and Lemma 5.10, we have that

$$\text{error}(\hat{w}) = \frac{\|\hat{w} - w\|^2}{n} \leq \frac{\tau^2}{ng_{\min}^2}\left(2\frac{\lambda_2 + 3\kappa}{\lambda_1} + \frac{4\mu_2}{\mu_1\tau^2}\right),$$

where $\kappa = \|U^tU - E^tE\|_2$. Plugging in this value, and the bounds on λ_2 and λ_1 from Lemma 5.6 and Lemma 5.8, we have that

$$\text{error}(\hat{w}) \leq \frac{\tau^2}{ng_{\min}^2}\left(\frac{3\mu_2 + 3D + 24\sqrt{\rho \log{(n/\delta)}}}{\mu_1\tau^2 - \mu_2} + \frac{4\mu_2}{\mu_1\tau^2}\right).$$

We simplify this by using $\mu_1\tau^2 - \mu_2 \geq \mu_1\tau^2/2$ to give

$$\begin{aligned} \text{error}(\hat{w}) &\leq \frac{\tau^2}{ng_{\min}^2}\left(\frac{6\mu_2 + 6D + 48\sqrt{\rho \log{(n/\delta)}}}{\mu_1\tau^2} + \frac{4\mu_2}{\mu_1\tau^2}\right) \\ &\leq \frac{10\tau^2}{ng_{\min}^2}\frac{1}{\mu_1\tau^2}\left(\mu_2 + D + 5\sqrt{\rho \log{(n/\delta)}}\right). \quad \square \end{aligned}$$

In order to illustrate our bounds better, we also state a corollary for (D, Δ)-regular graphs. This is also a restatement of Theorem 4.1 and thus completes its proof.

THEOREM 5.13 (THEOREM 4.1). *If G is a (D, Δ)-regular graph such that $\Delta > \frac{1}{8\epsilon\bar{w}^2}$, $D > \frac{256\log(n/\delta)}{\epsilon^2\bar{w}^2}$ and the second eigenvalue μ_2 satisfies the condition*

$$\mu_2 < \frac{\epsilon\bar{w}^2 D\Delta}{16},$$

then with probability $1 - \delta$, Algorithm 1 returns estimate $\hat{w}$, such that

$$\text{error}(\hat{w}) = O\left(\frac{\mu_2}{D\Delta} + \frac{1}{\Delta} + \frac{\sqrt{\log(n/\delta)}}{\sqrt{D}}\right) = O(\epsilon).$$

The proof is straightforward, after noting that $g_{\min} = \frac{1}{\sqrt{n}}$ and $nD = m\Delta$, and using a bound on $\mu_1 \geq \frac{m\Delta^2}{n}$, the average degree in G^tG.

Asymptotically, this gives $\text{error}(\hat{w}) = \tilde{O}(\frac{1}{\Delta} + \frac{1}{\sqrt{D}})$. Finally, we show that estimating the set of user reliabilities accurately enables us to estimate the quality of each item with small error. We show that for a random (D, Δ)-regular graph the total error in estimating item quality falls exponentially with the maximum item-degree, as well as with the average reliability. This is also a restatement of Theorem 4.2 and thus completes its proof.

THEOREM 5.14 (THEOREM 4.2). *Let G be a random (D, Δ)-regular graph. Let $\bar{w} = \sqrt{\frac{\sum_i w_i^2}{n}}$. Let $\hat{w}$ be an estimate with $error(w, \hat{w}) \leq \epsilon$. Then,* $\text{error}(\hat{q}) \leq e^{-\Delta(\bar{w}^2 - \epsilon)^2/64}$. *In particular, for $\hat{q}$ obtained by Algorithm 1,* $\text{error}(\hat{q}) \leq e^{-O(\Delta(\bar{w}^2 - \frac{1}{\Delta} - \frac{1}{\sqrt{D}})^2)}$.

The proof of this theorem is based on the following lemma.

LEMMA 5.15. *Denote $\alpha = \frac{\Delta}{n}(w \cdot \hat{w})$. Then (i) $\Delta \geq \alpha \geq \Delta(\bar{w}^2 - \epsilon)/2$ and (ii) if $\bar{w}^2 > \epsilon$, the probability that the ith item is wrong is at most $e^{-\alpha^2/16\Delta} \leq e^{-\Delta(\bar{w}^2 - \epsilon)^2/64}$.*

PROOF. For (i), note that $\epsilon = \text{error}(w, \hat{w}) = \|w - \hat{w}\|^2/n = \frac{|w|^2 + |\hat{w}|^2 - 2w\cdot\hat{w}}{n}$. Thus $w \cdot \hat{w}/n = (|w|^2 + |\hat{w}|^2 - n\epsilon)/2n \geq (\bar{w}^2 - \epsilon)/2$, which yields the result.

For (ii), define $z_i = \sum_j U_{ij}\hat{w}_j$. Then

$$\mathbb{E}[z_i] = \sum_j q_i\mathbb{E}[G_{ij}]w_j\hat{w}_j = q_i(\Delta/n)w \cdot \hat{w} = q_i\alpha.$$

Then from (i) and assuming $\bar{w}^2 > \epsilon$, we get $\alpha > 0$. Thus, $\text{sgn}(\mathbb{E}[z_i])$ is same as q_i. Thus $\text{sgn}(z_i) \neq q_i$ implies that $|z_i - \mathbb{E}[z_i]| > \mathbb{E}[z_i]$. Thus the probability that $\text{sgn}(z_i) \neq q_i$ is at most $\Pr[|z_i - \mathbb{E}[z_i]| > \mathbb{E}[z_i]]$.

For computing this probability, we will use Bernstein's inequality. Define $y_{ij} = U_{ij}\hat{w}_j$. Then $z_i = \sum_j y_{ij}$. Also $\mathbb{E}[y_{ij}] = q_i(\Delta/n)w_j\hat{w}_j$. Denote $x_{ij} = y_{ij} - \mathbb{E}[y_{ij}]$. Now we will apply Bernstein's inequality over x_{ij} for a fixed i but j from 1 to n. Note that $-1 - |\mathbb{E}[y_{ij}]| \leq x_{ij} \leq 1 + \mathbb{E}[y_{ij}]$. Thus, it is safe to say that $-2 \leq x_{ij} \leq 2$. Also

$$\mathbb{E}[x_{ij}^2] = \mathbb{E}[y_{ij}^2] - \mathbb{E}[y_{ij}]^2 = (\Delta/n)\hat{w}_j^2 - (\Delta/n)^2w_j^2\hat{w}_j^2.$$

Thus, $\mathbb{E}[x_{ij}^2] \leq (\Delta/n)\hat{w}_j^2(1 - (\Delta/n)w_j^2) \leq \Delta/n$. Applying Bernstein's inequality for $t = \alpha/2$, we get

$$\begin{aligned} \Pr\left[\left|\sum_j x_{ij}\right| \geq \alpha/2\right] &\leq e^{\frac{-\alpha^2/8}{\sum_j \mathbb{E}[x_{ij}^2] + 2(\alpha/2)(1/3)}} \\ &\leq e^{\frac{-\alpha^2/8}{\Delta + \alpha/3}} \leq e^{-\alpha^2/16\Delta}. \end{aligned}$$

Now $\sum_j x_{ij} = \sum_j y_{ij} - \mathbb{E}[\sum_j y_{ij}] = \sum_j y_{ij} - \mathbb{E}[z_i] = \sum_j y_{ij} - q_i\alpha$

Thus $|\sum_j y_{ij}| \geq |q_i\alpha| - |\sum_j x_{ij}| \geq \alpha - \alpha/2$ with probability $e^{-\alpha^2/16\Delta}$, which yields the result. $\square$

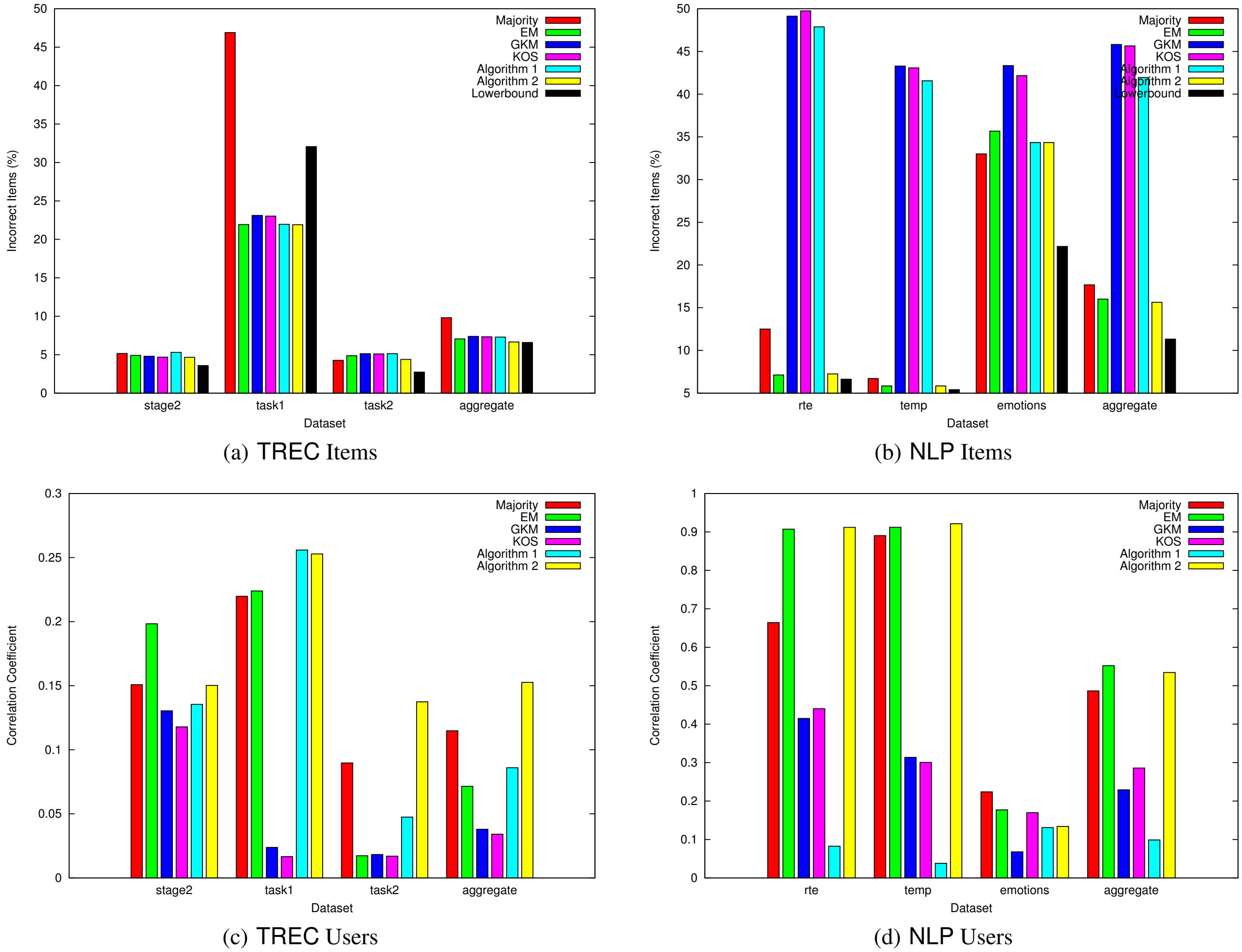
Analysis for Algorithm 2. The proof for Algorithm 2 follows a similar route. We first show a similar matrix concentration inequality and then use it to follow the the proof outline in Section 5. We postpone the details to the final version.

6. EXPERIMENTS

In this section we experimentally analyze the accuracy of the proposed algorithms in estimating both item ratings and user reliabilities. We implemented both Algorithm 1 and 2, which we denote by ALGORITHM 1 and ALGORITHM 2 respectively. We compare them with the following algorithms: the simple majority voting algorithm denoted by MAJORITY, the iterative EM algorithm denoted by EM, the spectral algorithm from Ghosh et al. [5] denoted by GKM, and the belief propagation algorithm from Karger et al. [10] denoted by KOS. We also implement LOWERBOUND which uses ground truth to compute the user reliabilities, and then uses the reliabilities to infer item ratings. Since it uses ground truth, it is not a true algorithm, but provides a benchmark to compare the performance of other algorithms.

Our implementation of ALGORITHM 1 and ALGORITHM 2 include the alternating projections heuristic described in Section 4.3.

Datasets. To illustrate the properties of our algorithms we use both synthetic and real datasets as described below.



(a) TREC Items

(b) NLP Items

(c) TREC Users

(d) NLP Users

Figure 1: Error analysis on real datasets: (a) and (b) measure error in item ratings estimates as % of incorrect items, and (c) and (d) measure error in user reliabilities using correlation coefficient. Lower % means better item estimates, while higher correlation coefficient means better user estimates. In all cases, ALGORITHM 2 is either best or second best. In terms of aggregate error, ALGORITHM 2 is best in both the item rating estimates and one user reliability estimate.

Name	m	labels	n	responses
TREC.stage2	3568	3568	181	10,751
TREC.task1	3297	3297	120	12000
TREC.task2	19033	2275	762	88385
NLP.rte	800	800	164	8000
NLP.temp	462	800	76	4620
NLP.emotions	600	600	228	6000

Table 1: Statistics for the real datasets used in our experiments.

(1) TREC[10]: this dataset is a collection of topic-document pairs labeled as relevant or non-relevant by mechanical turks. Several of the labels have ground truth assigned as well. There are three distinct datasets corresponding to different competitions of the workshop: namely, TREC.stage2, TREC.task1, and TREC.task2. The number of items, labeled items, users, and user responses for these datasets have been summarized in Table 1.

(2) NLP: this dataset [16] is a collection of three human judged datasets, all having ground truth labels, as summarized in Table 1.

(3) Synth: this is a synthetically generated dataset to help us analyze various algorithms in a controlled setting as a function of the numbers of responses by users and user reliabilities.

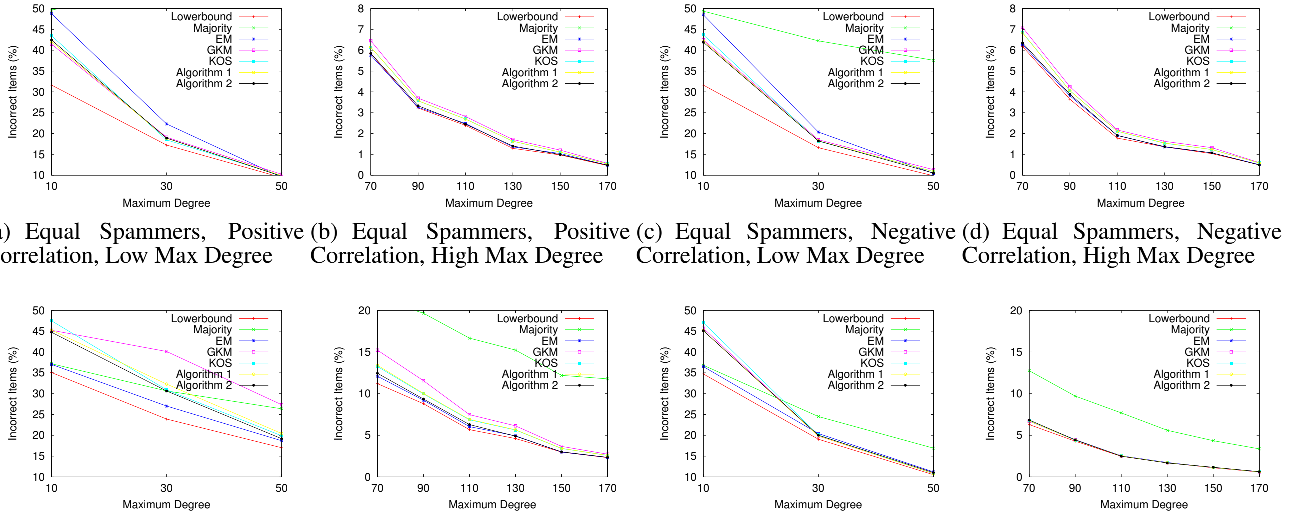
6.1 Real datasets

We compare the different algorithms over the TREC and NLP datasets. We evaluate both item rating estimates and user reliability estimates. Error in item ratings is measured in terms of % of incorrect item rating. Thus lower the value, better is the estimate.

Figure 1(a) shows the error for the three TREC datasets. We also show the overall aggregate error, which is the % of total items incorrectly predicted over the three datasets. For the first two datasets, the best algorithms are ALGORITHM 2 and EM, with MAJORITY much worst than the rest. This is perhaps because as we will see in synthetic datasets, MAJORITY is very sensitive to presence of spammers. In the third dataset, MAJORITY is in fact the best, along with ALGORITHM 2. Thus overall, ALGORITHM 2 is the most robust algorithm and has lowest aggregate error for the TREC dataset.[11]

[10] sites.google.com/site/treccrowd/home

[11] Surprisingly, LOWERBOUND for TREC.task1 is worse than some



(a) Equal Spammers, Positive Correlation, Low Max Degree

(b) Equal Spammers, Positive Correlation, High Max Degree

(c) Equal Spammers, Negative Correlation, Low Max Degree

(d) Equal Spammers, Negative Correlation, High Max Degree

(e) No Spammers, Positive Correlation, Low Max Degree

(f) No Spammers, Positive Correlation, High Max Degree

(g) No Spammers, Negative Correlation, Low Max Degree

(h) No Spammers, Negative Correlation, High Max Degree

Figure 2: Item errors on synthetic data. User degrees are drawn according to a power law. First row considers equal number of spammers, hammers, and random users, while second considers only hammers and random users. First two columns consider user reliabilities positively correlated with their degrees, while third and fourth considers negative correlation. We break each scenario into two graphs to better visualize the differences. In all graphs as max degree increases, so does the degree skew, and then ALGORITHM 2 **performs consistently better than** GKM **and** KOS**. In presence of spammers (first row),** MAJORITY **and** EM **deteriorate.** ALGORITHM 2 **performs well across the spectrum.**

Figure 1(b) shows the item errors for the three NLP datasets. Again we see a similar story here, ALGORITHM 2 is best in two out of the three datasets. For the third, MAJORITY is best, with ALGORITHM 2 not far behind. Overall, ALGORITHM 2 has the lowest aggregate error in NLP.

Next we analyze the error in user reliability estimates. Since some of the algorithms like KOS give user reliabilities only up to a constant normalization factor, we cannot directly measure user reliability estimates by comparing them to the ground truth (as they could be off by a constant factor). Thus we use Pearson's correlation coefficient to measure the accuracy of user reliability estimates, which is always a number between -1 and 1, and measures the correlation between two vector quantities. A value of 1 means complete positive correlation (up to some affine transformation), 0 means the two quantities are independent of each other, and -1 means they are negatively correlated. Larger the value, more the positive correlation, and therefore lower the error.

Figure 1(c) and 1(d) show that ALGORITHM 2 is either the best or close to the best in estimating user reliabilities for all the datasets, while other algorithms significantly underperform in at least one of the datasets.

6.2 Synthetic data

To better understand the performance of the algorithms with respect to the different parameters, we perform experiments over synthetic datasets. We generate synthetic datasets using the following steps. The number of items and the number of users is fixed to 1000 and 100 respectively. For the items, their binary ratings are generated as i.i.d. Bernoulli variables with $p = 1/2$.

For generating the bipartite graph between the items and users, we use powerlaw sequences for user degrees, where the number of items rated by users follow a powerlaw distribution with an exponent of 2.5. In each case, we generate a random graph satisfying the given degree sequence. We study the accuracy of different algorithms as a function of the maximum degree.

We define three types of users: hammers, which have reliability 0.8, spammers, who have reliability -0.8, and random, who have reliability of 0. We study the performance of algorithms as a function of the fraction of spammers, hammers and random users in the dataset. We consider two configurations: equal spammers, consisting of equal number of hammers, spammers and random users, and no spammers, consisting of equal number of hammers and random users.

To model real-life scenarios we consider cases when the user reliabilities are correlated with degrees. For e.g., reliable users could be more expensive, and hence offer less number of labels. Thus we consider the case of negative correlation where reliabilities are negatively correlated to the user degrees. For the sake of completeness, we also consider the case of positive correlation where reliabilities are positively correlated to user degrees.

This gives us four combinations: equal vs. no spammers and positive vs. negative correlations. Figure 2 shows the performance of all the algorithms for the four combinations. We explain the results below.

Figures 2(a) and 2(b) contains the results of the dataset with equal spammers and positive correlations. We break the graph into two parts to focus on the low and high degree parts separately. Because of a large number of spammers, MAJORITY has an error rate close to 50%, which is so large that it does not even appear in the plot. EM also has a very large error for low max degree, but becomes competitive for high max degree. As the maximum user degrees become larger, the skew in degrees also becomes larger, and

algorithms. This is because many of the users are close to random, as evident in high % errors for this dataset. Thus having a true estimate for these random users is not helpful, and LOWERBOUND is in fact worse than some of the other algorithms.

we notice that ALGORITHM 2 performs consistently better than the spectral methods of GKM and KOS for high maximum degree. This difference, although slight in synthetic data, manifests as a large one in real datasets, where the degree sequences are even more non-uniform. We see a very similar trend in Figures 2(c) and 2(d).

For Figures 2(e) and 2(f), which have no spammers and positive correlation, MAJORITY and EM do better than before. In fact, EM does slightly better than the spectral algorithms. Among the spectral algorithms, ALGORITHM 2 outperforms everyone else because of the non-uniform degree sequence. Figures 2(g) and 2(h) show as a similar trend for negative correlations as in the case of positive correlation, but the effect is less pronounced with all the algorithms bunched together more closely.

In summary, KOS and GKM perform well when the degrees are uniform (maximum degree is small and close to the minimum), but deteriorate when there is a skew in the degrees. EM performs well when there are no spammers, but deteriorates with the introduction of spammers. ALGORITHM 2 works well across the spectrum, and is robust to spammers and non-uniform degree sequences. This helps ALGORITHM 2 perform well on most synthetic and real datasets.

7. CONCLUSIONS

We studied the problem of aggregating user ratings when the user–item rating graph is arbitrary. We formulated a matrix completion problem and presented two eigenvector-based algorithms that have guaranteed error bounds when the resulting user–user corating graph satisfies expansion properties. It would be interesting to see if one can say anything directly about the alternate-projection based technique under a similar set of assumptions. In practice not all items need similar effort to rate; incorporating this difficulty is also an interesting open direction.

8. REFERENCES

[1] S. Boyd and J. Dattorro. Alternating projections, 2003. `www.stanford.edu/class/ee392o/altproj.pdf`.

[2] B. Carpenter. A hierarchical Bayesian model of crowdsourced relevance coding. In *Prof. 12th TREC*, 2011.

[3] A. Dawid and A. Skene. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Applied Statistics*, pages 20–28, 1979.

[4] O. Dekel and O. Shamir. Vox populi: Collecting high-quality labels from a crowd. In *Proc. 22nd COLT*, 2009.

[5] A. Ghosh, S. Kale, and R. P. McAfee. Who moderates the moderators?: Crowdsourcing abuse detection in user-generated content. In *Proc. 12th EC*, pages 167–176, 2011.

[6] A. Ghosh and R. P. McAfee. Crowdsourcing with endogenous entry. In *Proc. 21st WWW*, pages 999–1008, 2012.

[7] N. Gillis and F. Glineur. Low-rank matrix approximation with weights or missing data is NP-hard. *SIAM J. Matrix Analysis Applications*, 32(4):1149–1165, 2011.

[8] P. G. Ipeirotis and P. K. Paritosh. Managing crowdsourced human computation: a tutorial. In *Proc. 20th WWW (Companion Volume)*, pages 287–288, 2011.

[9] E. Kamar and E. Horvitz. Incentives for truthful reporting in crowdsourcing. In *AAMAS*, pages 1329–1330, 2012.

[10] D. Karger, S. Oh, and D. Shah. Iterative learning for reliable crowdsourcing systems. In *Proc. 25th NIPS*, pages 1953–1961, 2011.

[11] Q. Liu, J. Peng, and A. Ihler. Variational inference for crowdsourcing. In *Proc. 26th NIPS*, 2012.

[12] P. K. Paritosh, P. Ipeirotis, M. Cooper, and S. Suri. The computer is the new sewing machine: benefits and perils of crowdsourcing. In *Proc. 20th WWW (Companion Volume)*, pages 325–326, 2011.

[13] V. C. Raykar and S. Yu. Eliminating spammers and ranking annotators for crowdsourced labeling tasks. *JMLR*, 13:491–518, 2012.

[14] V. C. Raykar, S. Yu, L. H. Zhao, G. H. Valadez, C. Florin, L. Bogoni, and L. Moy. Learning from crowds. *JMLR*, 11:1297–1322, 2010.

[15] V. Sheng, F. Provost, and P. Ipeirotis. Get another label? Improving data quality and data mining using multiple, noisy labelers. In *Proc. 14th KDD*, pages 614–622, 2008.

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

[17] J. A. Tropp. User-friendly tail bounds for sums of random matrices. *Found. Comput. Math.*, 12:389–434, 2012.

[18] P. Welinder, S. Branson, S. Belongie, and P. Perona. The multidimensional wisdom of crowds. In *Proc. 24th NIPS*, pages 2424–2432, 2010.

[19] D. Zhou, J. Platt, S. Basu, and Y. Mao. Learning from the wisdom of crowds by minimax entropy. In *Proc. 26th NIPS*, pages 2204–2212, 2012.