Journal of Machine Learning Research 21 (2020) 1-36 Submitted 5/19; Revised 12/19; Published 6/20

Gradient Descent for Sparse Rank-One Matrix Completion for Crowd-Sourced Aggregation of Sparsely Interacting Workers[*]

Yao Ma YAOMA@BU.EDU
Division of Systems Engineering
Boston University

Alex Olshevsky ALEXOLS@BU.EDU
Department Electrical and Computer Engineering
Division of Systems Engineering
Boston University

Venkatesh Saligrama SRV@BU.EDU
Department Electrical and Computer Engineering
Boston University

Csaba Szepesvari SZEPI@GOOGLE.COM
Google Deepmind

Editor: Inderjit Dhillon

Abstract

We consider worker skill estimation for the single-coin Dawid-Skene crowdsourcing model. In practice, skill-estimation is challenging because worker assignments are sparse and irregular due to the arbitrary and uncontrolled availability of workers. We formulate skill estimation as a rank-one correlation-matrix completion problem, where the observed components correspond to *observed* label correlation between workers. We show that the correlation matrix can be successfully recovered and skills are identifiable if and only if the sampling matrix (observed components) does not have a bipartite connected component. We then propose a projected gradient descent scheme and show that skill estimates converge to the desired global optima for such sampling matrices. Our proof is original and the results are surprising in light of the fact that even the weighted rank-one matrix factorization problem is NP-hard in general. Next, we derive sample complexity bounds in terms of spectral properties of the *signless* Laplacian of the sampling matrix. Our proposed scheme achieves state-of-art performance on a number of real-world datasets.

Keywords: distributed optimization, stochastic gradient descent

[*] This work was supported partly by the National Science Foundation Grant 1527618, the Office of Naval Research Grant N0014-18-1-2257, the Hariri Institute's Data Science Fellowship and by a gift from the ARM corporation.

1. Introduction

Crowdsourcing can be a scalable approach to collecting data for tasks that require human knowledge such as image recognition and natural language processing. Through crowdsourcing platforms such as Amazon Mechanical Turk, a large number of data tasks can be assigned to workers who are asked to give binary or multi-class labels. The goal of much of crowdsourcing research is to estimate the unknown ground truth, given that the quality of the workers can be variable. Indeed, due to the high variability of worker skills, aggregating true labels becomes a challenging problem.

One straightforward approach is to directly estimate the unknown labels by majority voting from the information provided by workers. In this approach, an implicit assumption is that all workers have identical skills on each task; on the other hand, one might expect the answers from reliable workers are more likely to be accurate. In practice, the crowd is often highly heterogeneous in terms of skill levels, and downweighting unskilled workers and upweighting skilled workers can have a significant impact on the performance. Many aggregation methods ranging from the weighted majority vote to more complex schemes that incorporate worker quality and accuracy have been proposed. Theoretically, recent works (Berend and Kontorovich, 2014; Szepesvári, 2015) have investigated the importance of having precise knowledge of skill quality for accurate prediction of ground-truth labels. Moreover, accurate skill estimation can also be useful for other purposes like worker training, task assignment, or for use in worker-compensation schemes.

There are two challenges in estimating skills of workers given that the problem setup is unsupervised. The first challenge is to construct a skill model for each worker. Many papers achieve empirical success by applying Dawid & Skene (DS) model (Dawid and Skene, 1979), which is a simple model that parameterized by the probability of a worker answers the true label. In this paper, the basis of our works is the homogeneous DS model where each worker is assumed to have the same skill level on each class. More specifically, we focus on the single-coin (DS) model for binary crowdsourcing problem in this paper (though in Section 4, we extend our algorithm to multiclass problems).

The second challenge is that, in practice, workers are only available for a short period of time which means only a small subset of data is labeled by each worker. This introduces a sparse worker-task assignment (Karger et al., 2013; Dalvi et al., 2013). An additional subtle issue is the lack of diversity in terms of interactions between the workers: a worker is often grouped with a limited subset of workers across all tasks. This situation is remarkably evident on benchmark datasets: The 'Web' dataset has 177 workers, with 3 to 20 workers/task and each worker on average interacting with about another 2.7 workers only, while the standard deviation of how many workers a worker is interacting with is 15. The 'RTE' dataset has 164 workers, has only 10 workers/task on the average and each worker interacts with fewer than 2.5 other workers, while the standard deviation of the interaction degree is 20. This is in contrast to most existing crowdsourcing research which only considered estimate skills with nearly complete data. We are therefore motivated by the need to make spectral methods suitable for non-regular worker-task data often seen in practice.

In this paper, we suppose that the input comes in the form of a *sparsely filled* $W \times T$ worker-task label matrix. The workers possess unique unknown skills, and tasks assume unique unknown labels. The worker-task label matrix collects the random labels provided by the workers for individual tasks. The skill level of a worker is the (scaled) probability of the worker's label matching the true unknown label for any of the tasks. The observed labels are independent of each other.

Given the workers' skill levels, the optimal way (Nitzan and Paroush, 1981; Shapley and Grofman, 1984) to reconstruct the unknown labels is to use weighted majority voting where the weights assigned to the label provided by a worker is equal to the log-odds underlying the worker's skill. Since skill levels are unknown, we follow prior works (Dalvi et al., 2013; Berend and Kontorovich, 2014; Szepesvári, 2015; Bonald and Combes, 2016) and adopt a two-step approach, whereby worker skills are first estimated and then these skills are used with the optimal weighting method to recover labels. Our main contributions are as follows:

1. We construct a skill estimator under single-coin model as a weighted least-squares rank-one matrix completion/factorization problem. The matrix being factored is the correlation matrix among the workers, with the weights compensating for the varying accuracy in the inter-worker correlations.

2. We show that skills can be recovered from the observation data matrix whenever the worker-worker interaction graph does not contain a bipartite connected component. In particular, for any crowdsourcing problem that has non-bipartite worker-worker interaction graph, there always exists a method to estimate true skills.

3. In the context of minimizing the objective function, we propose to use projected gradient descent which is theoretically verified to converge to the true skills. We give natural and mild conditions on the weighting matrix under which we prove that projected gradient descent, despite the objective being non-convex, is guaranteed to find the rank-one decomposition of the true moment matrix.

4. We extend our algorithm to multiclass case by applying the homogeneous DS model. Under this model, we prove that any multiclass problem can be formulated as a weighted least-squares rank-one problem where the unknown variable is a linear function of true skills.

Our approach is also of independent interest, as we derive a fundamental result about symmetric rank-one matrix completion: the unobserved entries can be recovered by gradient descent in polynomial time whenever the sampling matrix is irreducible and non-bipartite. Our results for convergence of the proposed gradient descent scheme should be somewhat surprising given that the related weighted low-rank factorization problem is known to be NP-hard even for the rank-one case (Gillis and Glineur, 2011). In contrast to our approach, existing results in low-rank matrix completion require strong assumptions on the weighting matrix, typically some form of incoherence, e.g., (Ge et al., 2016).

2. Related Work

Discriminative Approach: In contrast to our two-step approach, several works adopt a discriminative method for label prediction. Specifically, Li and Yu (2014); Tian and Zhu (2015) directly identify true labels by various aggregation rules that incorporate worker reliability.

Skill Estimation: As mentioned earlier, we work in the problem of estimating skills under the single-coin model. Past approaches to skill estimation are based on *maximum likelihood/maximum posteriori* (ML/MAP) estimation, or *moment matching*, or a combination of these. In particular, various versions of the EM algorithm have been proposed to implement ML/MAP estimation, starting with the work of Dawid and Skene (1979). Variants and extensions of this method, tested in various problems, include Hui and Walter (1980); Smyth et al. (1995); Albert and Dodd (2004); Raykar et al. (2010); Liu et al. (2012). A number of recent works were concerned with performance guarantees for Expectation Maximization (EM) and some of its variants (Gao and Zhou, 2013; Zhang et al., 2014; Gao et al., 2016). Another popular direction is to add priors over worker skills, labels or worker-task assignments. To properly deal with the extra information, various Bayesian methods (belief propagation, mean-field and variational methods) have been considered (Raykar et al., 2010; Karger et al., 2011; Liu et al., 2012; Karger et al., 2013, 2014). Moment matching is also widely used (Ghosh et al., 2011; Dalvi et al., 2013; Zhang et al., 2014; Gao et al., 2016; Bonald and Combes, 2016; Zhang et al., 2016). With the exception of Bonald and Combes (2016), who propose an ad-hoc method, the algorithms in these works use matrix or tensor factorization.[1]

In theory, an ML/MAP method which is *guaranteed* to maximize the likelihood/posterior, is the ideal method to accommodate irregular worker-task assignments. However, as far as we know, none of the existing algorithms, unless initialized with a moment-matching-based spectral method, is proven to indeed find a satisfactory approximate maximizer of the objective that it is maximizing (Zhang et al., 2016). At the same time, moment matching methods that use spectral (and in general algebraic) algorithms implicitly assume the regularity of worker-task assignments, too. Indeed, the approach of Ghosh et al. (2011) crucially relies on the regularity of the worker-task assignment (as the method proposed uses unnormalized statistics). In particular, this method is not expected to work at all on non-regular data. Other spectral methods, being purely algebraic, implicitly treat all entries in the estimated matrices and tensors as if they had the same accuracy, which, in the case of irregular worker-task assignments, is far from the truth. In particular, the need to explicitly deal with data with unequal accuracy is a widely recognized issue that has a long history in the low-rank factorization community, going back to the work of Gabriel and Zamir (1979). Starting with this work, the standard recommendation is to reformulate the low-rank estimation problem as a weighted least-squares problem (Gabriel and Zamir, 1979; Srebro and Jaakkola, 2003). In this paper, we will also follow this recommendation.

While Dalvi et al. (2013) also use a weighted least-squares objective, this is not by choice, but rather as a consequence of the need to normalize the data rather than to correct for the inaccuracy of the data. Furthermore, rather than considering the direct minimization of the resulting objective, they use two heuristic approaches that also use an unweighted spectral method.

In this light, our goal is to make spectral methods suitable for non-regular worker-task data often seen in practice.

Matrix Factorization/Completion: Unlike the general matrix factorization problem arising in recommender systems (Koren et al., 2009), we are primarily concerned with a rank-one

[1] While Ghosh et al. (2011) pioneered the matrix factorization approach, their work is less relevant to this discussion as they estimate the labels directly.

estimation of square symmetric matrices. Existing results on matrix completion (Ge et al., 2016) for square symmetric matrices are more general but require stronger assumptions on the matrix such as incoherence and random sampling.

Notation and conventions: The set of reals is denoted by $\mathbb{R}$, the set of natural numbers which does not include zero is denoted by $\mathbb{N}$. For $k \in \mathbb{N}$, $[k] \doteq \{1, \ldots, k\}$. Empty sums are defined as zero. We will use $\mathbb{P}$ to denote the probability measure over the measure space holding our random variables, while $\mathbb{E}$ will be used to denote the corresponding expectation operator. For $p \in [1, \infty]$, we use $\|v\|_p$ to denote the p-norm of vectors. Further, $\|\cdot\|$ stands for the 2-norm, $\|\cdot\|_{\mathrm{F}}$ is the Frobenius-norm. The cardinality of a set S is denoted by $|S|$. For a real-valued vector x, $|x|$ denotes the vector whose ith component is $|x_i|$. Proofs of new results, missing from the main text are given in the appendix.

3. Formal problem statement

We first consider binary crowdsourcing tasks where a set of workers provide $\{-1, 1\}$ labels for a large number of items. Let $W \in \mathbb{N}$ be a positive integer denoting the number of workers. A problem instance $\theta \doteq (s, A, g)$ is given by: a skill vector $s = (s_1, \ldots, s_W) \in [-1, 1]^W$ associating the skill level s_w with worker w; the worker-task assignment set $A \subset [W] \times \mathbb{N}$, which captures which workers provide labels on which tasks; and the vector of "ground truth labels" $g \in \{\pm 1\}^{\mathbb{N}}$, which are unknown and which we would like to estimate.

When $A \subset [W] \times [T]$ for some $T \in \mathbb{N}$, we say that θ is a finite instance with T tasks; otherwise we will say θ is an infinite instance. We allow infinite tasks to be able to discuss asymptotic identifiability.

It will be convenient to use Θ_W to denote the set of all possible problem instances, defined as above. For any instance $\theta \in \Theta_W$, the worker-task assignment set provides important information about worker interaction structure. Indeed, we can think of two workers as "interacting" if they provide a label for the same task. Formally, we define the interaction graph as follows.

Definition 1 (Interaction Graph) *Let A be a worker-task assignment set. The (worker) interaction graph underlying A is an undirected graph $G = G_A$ with vertex set $[W]$ such that $G = ([W], E)$ with $(i, j) \in E$ if there exists some task $t \in \mathbb{N}$ such that both (i, t) and (j, t) are elements of A.*

In the case of infinite instance, the interaction graph is a unweighted graph where an edge does not have any weight associated with it. For finite instances, it will make sense to assign a weight on the edge (i, j) which is the number of tasks shared by workers i and j.

Our goal is to recover the ground truth labels $(g_t)_t$ given observations $(Y_{w,t})_{(w,t) \in A}$, where $Y_{w,t}$ is a ± 1 random variable associated with worker w and task t. According to the single-coin model, the observations are generated as $Y_{w,t} = Z_{w,t} g_t$, where $(Z_{w,t})_{(w,t) \in A}$ is a collection of mutually independent random variables that satisfy $\mathbb{E}[Z_{w,t}] = s_w$. Note that this is the same as assuming that worker w returns g_t with probability $(1 + s_w)/2$ and $-g_t$ with probability $(1 - s_w)/2$.

Thus a worker w with $s_w = 1$ always returns the ground truth, while a worker with $s_w = -1$ always return the opposite label of the ground truth; and a worker with $s_w = 0$

will always provide a random variable $Y_{w,t}$ with zero expectation regardless of the ground truth, i.e., a uniformly random label.

Remark: As will be discussed in detail later, some additional assumptions on the skill vector s will be needed for accurate estimation of ground truth labels; obviously, little can be done if s is the zero vector, i.e., if every worker returns uniformly random labels irrespective of the ground truth. Another obvious observation is, in the event that all workers agree, we cannot distinguish the possibility that s is proportional to the all ones vector (and every worker provides the right label) from the possibility that s is proportional to the negative of the all ones vector (and every worker provides the wrong label). One way to get around this problem is to assume that $s > 0$, i.e., all workers have at least some skill; we will make this assumption when analyzing our projected gradient descent method. A weaker approach is to assume that $\sum_w s_w > 0$, i.e., that, on the net, the workers are collectively more prone to return correct rather than incorrect labels; as we discuss later, this is sufficient for identifiability.

A (deterministic) *inference method* underlying an assignment set A takes the observations $(Y_{w,t})_{(w,t)\in A}$ and returns a real-valued score for each task in A; the signs of the scores give the label-estimates. Inference methods are aimed at working with finite assignment sets. To process an infinite assignment set, we define the notion of *inference schema.* In particular, an *inference schema* underlying an infinite assignment set A is defined as the infinite sequence of inference methods $\gamma^{(1)}, \gamma^{(2)}, \dots$ such that $\gamma^{(t)}$ is an inference method for the first t tasks.

When important, we will use the subindex θ in $\mathbb{P}_\theta$ to denote the probability when the problem instance is θ. We will use $\mathbb{E}_\theta$ to denote the corresponding expectation operator. With this notation, the expected loss suffered by an inference schema $\gamma = (\gamma^{(1)}, \gamma^{(2)}, \dots)$ on the first T tasks of an instance θ is

$$\mathcal{L}_T(\gamma;\theta) = \frac{1}{T}\,\mathbb{E}_\theta\Big[\textstyle\sum_{t=1}^T \mathbb{I}\left\{\gamma_t^{(T)}(Y) g_t \le 0\right\}\Big]\,.$$

The optimal inference schema for an assignment set A *given* the knowledge of the skill vector $s \in [-1,1]^W$ is denoted by $\gamma^*_{s,A}$. The next section gives a simple explicit form for this optimal schema. The *average regret* of an inference schema $\gamma = (\gamma^{(1)}, \gamma^{(2)}, \dots)$ for an instance $\theta \in \Theta$ is its excess loss on the instance θ as compared to the loss of the optimal schema:

$$\overline{R}_T(\gamma;\theta) = \mathcal{L}_T(\gamma;\theta) - \mathcal{L}_T(\gamma^*_{s,A};\theta)\,.$$

If the average regret converges to zero, then the loss suffered by γ asymptotically converges to the loss of the optimal inference. Based on this, we define asymptotic consistency and learnability:

Definition 2 (Consistency and Learnability) *An inference schema is said to be (asymptotically)* consistent *for an instance set $\Theta \subset \Theta_W$ if, for any $\theta \in \Theta$,* $\limsup_{T\to\infty} \overline{R}_T(\gamma) = 0$. *An instance set $\Theta \subset \Theta_W$ is (asymptotically)* learnable *if there is a consistent inference schema for it.*

3.1. Two-Step Plug-in Approach

In this work we will pursue a two-step approach based on first estimating the skill vector s and then utilizing a plug-in classifier to predict the ground-truth labels. The motivation for a two-step approach stems from existing results that characterize accuracy in terms of skill estimation errors. For the sake of exposition, we recall some of these results now.

It has been shown in Li and Yu (2014), the optimal classifier is log-odds weighted majority voting given by the MAP rule. Suppose the prior distribution of true labels is the uniform distribution over $\{-1, 1\}$; then the Bayes classifier is well-known to be the optimal classifier (Duda et al., 2001),i.e.,

$$\begin{aligned}\gamma^*_{s,A}(Y_t) &= \mathrm{argmax}_{l\in\{+1,-1\}}\, \mathbb{P}(g_t = l|Y_t, s, A)\\ &= \mathrm{argmax}_{l\in\{+1,-1\}}\, \mathbb{P}(g_t = l)\mathbb{P}(Y_t|g_t, s, A)\\ &= \mathrm{argmax}_{l\in\{+1,-1\}}\, \log \mathbb{P}(Y_t|g_t, s, A)\\ &= \mathrm{argmax}_{l\in\{+1,-1\}} \sum_{i=1}^{W} \log \frac{1+s_i}{1-s_i} \mathbb{I}\left\{Y_{i,t} = l\right\}, \end{aligned} \tag{1}$$

where the third equation follows the assumption that $\mathbb{P}(g_t = +1) = \mathbb{P}(g_t = -1) = 1/2$, and the fourth equation, after some algebra, is compact write to write the MAP estimator. Notice that $\gamma^*_{s,A}$ is a function of only one parameter, namely the skill vector s.

Regarding the loss of the optimal schema, we start with introducing result when skills are known in advance. In this case, Berend and Kontorovich (2014) provides an upper error bound, as well as an asymptotically matching lower bound, which are stated as follows:

Lemma 3 *For any task $t \in \mathbb{N}$, the optimal decision rule γ^* satisfies*

$$\begin{aligned}\mathbb{P}\left(\gamma^*_t(Y) \neq g_t\right) &\leq \exp\left(-\frac{1}{2}\Phi\right),\\ \mathbb{P}\left(\gamma^*_t(Y) \neq g_t\right) &\geq \frac{3}{4[1+\exp\left(2\Phi + 4\sqrt{\Phi}\right)]},\end{aligned}$$

where $\Phi = \sum_{i\in W} s_i \log\left(\frac{1+s_i}{1-s_i}\right)$ is called committee potential.

However, we do not assume that we know the skills of workers in reality, and thus the true optimal inference classifier is unknown to us. One natural way is to construct a true label inference that approximates the optimal Bayes classifier via estimating workers' skills. Fortunately, in addition to the case of known skills, Szepesvári (2015); Berend and Kontorovich (2014) also provide an error bound when skills are only estimated:

Lemma 4 *For any $\epsilon > 0$, the loss with estimated weights $\hat{v}_i = v(\hat{s}_i)$ satisfies*

$$\begin{aligned}&\frac{1}{T}\,\mathbb{E}_\theta\Big[\sum_{t=1}^{T} \mathbb{I}\left\{\gamma_{t,\hat{s}}(Y) g_t \leq 0\right\}\Big]\\ &\leq \frac{1}{T}\,\mathbb{E}_\theta\Big[\sum_{t=1}^{T} \mathbb{I}\left\{\gamma^*(Y) g_t \leq \epsilon\right\}\Big] + \mathbb{P}_\theta(\|v^* - \hat{v}\|_1 \geq \epsilon)\,.\end{aligned}$$

In turn, the error $\|v^* - \hat{v}\|_1$ can be bounded in terms of the multiplicative norm-differences in the skill estimates (see Berend and Kontorovich (2014)):

Lemma 5 *Suppose* $\frac{1+\hat{s}_i}{1+s_i}, \frac{1-\hat{s}_i}{1-s_i} \in [1-\delta_i, 1+\delta_i]$ *then* $|v(s_i) - v(\hat{s}_i)| \leq 2|\delta_i|$.

These results together imply that a plug-in estimator with a guaranteed accuracy on the skill levels, in turn, leads to a bound on the error probability of predicting ground-truth labels. This motivates the skill estimation problem, which we consider in the remainder of this paper.

4. Weighted Least-Squares Estimation

In this section, we propose an asymptotically consistent skill estimator for potentially sparse worker-task assignments. We are motivated by the scenario when, for most workers, only a very small portion of tasks are assigned to them. This induces not only an extremely sparse worker-task assignment graph, but more importantly a sparse worker-worker interaction graph.

Recall that given a problem instance $\theta = (s, A, g)$, the data of the learner is given by the matrix $(Y_{i,t})_{(i,t)\in A}$ which is a collection of independent binary random variables such that $Y_{i,t} = g_t Z_{i,t}$ and $s_i = \mathbb{E}(Z_{i,t})$. When A is finite, we define $N \in \mathbb{N}^{W\times W}$ to be the matrix whose (i,j)th entry with $i \neq j$ gives the number of times the workers i and j labeled the same task:

$$N_{ij} = |\{t \in \mathbb{N} : (i,t), (j,t) \in A\}|$$

and we also let $N_{ii} = 0, \forall i = 1, \ldots, W$. Note that there is an edge between workers i and j in the interaction graph exactly when $N_{ij} > 0$.

When A is infinite, N_{ij} may be infinite. In this case, for $i \neq j$ we also define $N_{ij}(T) = |\{t \in [T] : (i,t), (j,t) \in A\}|$ to denote the number of times workers i and j provide a label for the same task in the first T tasks, and similarly we let $N_{ii}(T) = 0$ for all i.

The starting point of our approach is the following observation about the single coin model: the expected correlation between each pair of workers is ground-truth independent. Indeed,

$$\begin{aligned}
\mathbb{E}[Y_{i,t}Y_{j,t}] &= \mathbb{E}[g_t Z_{i,t} g_t Z_{j,t}] \\
&= \mathbb{E}[Z_{i,t} Z_{j,t}] \\
&= \left(\frac{1+s_i}{2}\frac{1+s_j}{2} + \frac{1-s_i}{2}\frac{1-s_j}{2}\right)\cdot 1 \\
&\quad + \left(\frac{1+s_i}{2}\frac{1-s_j}{2} + \frac{1-s_i}{2}\frac{1+s_j}{2}\right)\cdot(-1) \\
&= s_i s_j,
\end{aligned}$$

where the second equation used that $g_t^2 = 1$.

This observation motivates estimating the skills using

$$\tilde{s} = \operatorname{argmin}_{x\in[-1,+1]^W} \frac{1}{2} \sum_{i,j,t \;|\; (i,t),(j,t)\in A} (Y_{i,t}Y_{j,t} - x_i x_j)^2 \tag{2}$$

Note that the number of terms containing the skill estimate x_i of particular worker i in this objective scales with how many other workers this worker i works with. Intuitively, this should feel "right": the more a worker works with others, the more information we should have about its skill level.

As it turns out, there is an alternative form for this objective, which is also very instrumental and which will form the basis of our algorithm and also of our analysis. To introduce this form, define $C_{ij} \doteq s_i s_j$ and let its empirical estimation be

$$\tilde{C}_{ij} = \frac{1}{N_{ij}} \textstyle\sum_{t \,\mid\, (i,t),(j,t) \in A} Y_{i,t} Y_{j,t} \,. \tag{3}$$

An alternative form of the objective in Eq. (2) is given by the following result:

Lemma 6 *Let $L : [-1,1]^W \to [0,\infty)$ be defined by*

$$L(x) = \frac{1}{2} \sum_{(i,j)\in E} N_{ij} (\tilde{C}_{ij} - x_i x_j)^2.$$

The optimization problem of Eq. (2) *is equivalent to the optimization problem*

$$\operatorname{argmin}_{x\in[-1,+1]^W} L(x).$$

The proof, which is just simple algebra to show the two objective functions are equal up to a constant shift, is given in Appendix A.

The objective function from Lemma 6 can be seen as a weighted low-rank objective, first proposed by Gabriel and Zamir (1979). Clearly, the objective prescribes to approximate $\tilde{C}$ using $xx^\top$, with the error in the (i,j)th entry scaled by N_{ij}. Note that this weighting is reasonable as the variance of $\tilde{C}_{ij}$ is proportional to $1/N_{ij}$ and we expect from the theory of least-squares that an objective combining multiple terms where the data is heteroscedastic (has unequal variance), the terms should be weighted with the inverse of the data variances. Since $N_{ii} = 0$, the weighting function N can in general be full-rank, and in this case the general weighted rank-one optimization approximation known to be NP-hard (Gillis and Glineur, 2011).

However, our data has special structure, which will allow one to avoid the existing hardness results. Indeed, on the one hand, as the number of data points increases, $\tilde{C}_{ij}$ will be near rank-one itself; and, on the other hand, we will put natural restrictions on the weighting matrix which are in fact necessary for identifiability. These conditions will allow us to avoid the NP-hardness results of Gillis and Glineur (2011).

4.1. Plug-in Gradient Descent

To solve the weighted least-squares objective, the simplest algorithm is the gradient descent algorithm. We propose a *Plug-in Gradient Descent* (PGD) algorithm that sequentially updates the skill level based on following the (negative) gradient of the loss L at each time step:

$$\tilde{x}_i^{t+1} = x_i^t + \eta \sum_{(i,j)\in E} N_{ij} (\tilde{C}_{ij} - x_i^t x_j^t) x_j^t$$

where $N_i = |\{t : (i,t) \in A\}| = \sum_{j=1}^{W} N_{ij}$ is the number of tasks labeled by worker i and $\tau > 0$ is a tuning parameter. We do not necessarily need to explicitly enforce the constraint that $x \in [-1,1]^n$, though we have found that it helps in terms of the practical performance of the method, as we'll remark in Section 6 where we discuss experimental results.

4.2. An Extension to Multi-class Classification

We now briefly describe how our approach may be extended to the case when the labels are not binary. Above, we have shown how the binary case may be reduced to a noisy rank-one matrix completion problem as in Lemma 6. Here we show how the same approach can be used for the multiclass case.

As before, we suppose that $W \in \mathbb{N}$ workers are asked to provide labels to a series of M-class classification tasks whose ground truths $g_t, t = 1, \ldots, T$ are unknown. We will use a one-hot encoding of the ground truths, i.e., $g_t \in \mathbb{R}^M$ will be expressed as $g_t \in \{[1,0,0,\ldots,0]^T, [0,1,0\ldots,0]^T, \ldots, [0,0,\ldots,0,1]^T\} \in \mathbb{R}^M$.

We will associate a skill level with every worker using a homogeneous Dawid-Skene model, where each worker is assumed to have the same accuracy and error probabilities on each class. Formally, worker i provides label $l \in \mathbb{R}^M$ with probability

$$\begin{cases} \mathbb{P}(Y_{i,t} = l)) = p_i & \text{if } l = g_t \\ \mathbb{P}(Y_{i,t} = l)) = \frac{1-p_i}{M-1} & \text{if } l \neq g_t. \end{cases}$$

Similar to binary tasks, Li and Yu (2014) showed that the optimal prediction method under homogeneous Dawid-Skene model is weighted majority voting. More specifically, when $p_i, \forall i = 1, \ldots, W$ are known, the oracle MAP rule is

$$\gamma^*_{s,A}(Y) = \text{argmax}_{l \in [M]} \sum_{i:(i,t)\in A} v_i^* \mathbb{I}\left\{Y_{i,t} = l\right\},$$

where $v_i^* = \log \frac{(M-1)p_i}{1-p_i}, \forall i \in [W]$. The proof of this can be obtained by following the same line as in Section 3.1.

In order to construct the weighted majority voting model, we extend PGD algorithm to handle multi-class tasks by showing the skill estimation problem is still a rank one matrix completion problem as follows.

Lemma 7 *Let us define skill levels*

$$s_i = \frac{M}{M-1} p_i - \frac{1}{M-1},$$

and noisy covariances

$$\tilde{C}_{ij} = \frac{1}{N_{ij}} \sum_{t \ | \ (i,t),(j,t)\in A} \langle Y_{i,t}, Y_{j,t} \rangle.$$

Then

$$E\left[\frac{M-1}{M}\tilde{C} - \frac{1}{M-1}\right] = ss^T.$$

Proof Since the random vectors $Y_{i,t}$ and $Y_{j,t}$ are independent, we can write the expectation of the inner product of $Y_{i,t}$ and $Y_{j,t}$ as

$$\mathbb{E}[\langle Y_{i,t}, Y_{j,t}\rangle] = p_i p_j + \frac{1}{M-1}(1-p_i)(1-p_j).$$

This follows because the inner product is one only if $Y_{i,t} = Y_{j,t} = l$, for some label l, and the probability of this is either $p_i p_j$ or $\frac{1-p_i}{M-1}\frac{1-p_j}{M-1}$ depending on whether $l = g_t$ or $l \neq g_t$.

A simple algebraic manipulation gives the following

$$\mathbb{E}[\langle Y_{i,t}, Y_{j,t}\rangle] = \frac{M-1}{M}\left(\frac{M}{M-1}p_i - \frac{1}{M-1}\right)\left(\frac{M}{M-1}p_j - \frac{1}{M-1}\right) + \frac{1}{M}.$$

which implies

$$E\left[\frac{M}{M-1}\langle Y_{i,t}, Y_{j,t}\rangle - \frac{1}{M-1}\right] = ss^T.$$

We thus have

$$\begin{aligned}
E\left[\frac{M-1}{M}\tilde{C} - \frac{1}{M-1}\right] &= \frac{M}{M-1}\frac{1}{N_{ij}}\sum_{t \ | \ (i,t),(j,t)\in A} E\langle Y_{i,t}, Y_{j,t}\rangle - \frac{1}{M-1} \\
&= \frac{1}{N_{ij}}\sum_{t \ | \ (i,t),(j,t)\in A}\left(\frac{M}{M-1}E\langle Y_{i,t}, Y_{j,t}\rangle - \frac{1}{M-1}\right) \\
&= \frac{1}{N_{ij}}\sum_{t \ | \ (i,t),(j,t)\in A} ss^T \\
&= ss^T.
\end{aligned}$$

■

As a consequence of this lemma, if we define

$$\hat{C}_{ij} = \frac{M}{M-1}\frac{1}{N_{ij}}\sum_{t \ | \ (i,t),(j,t)\in A}\langle Y_{i,t}, Y_{j,t}\rangle - \frac{1}{M-1},$$

then s can be estimated by solving a rank one matrix completion problem with objective function

$$\tilde{s} = \operatorname{argmin}_{x\in[-\frac{1}{M-1},1]^W}\sum_{(i,j)\in E}(\hat{C}_{ij} - x_i x_j)^2.$$

As previously, in the limit as $t \to \infty$, the rank-one problem is an exact match for the problem of recovering skills. In the case where t is finite, we will be in the "noisy" regime where $\hat{C}$ can be thought of as a noise-corrupted version of the true rank-one matrix ss^T, with the amount of noise ill decaying to zero as $t \to \infty$.

5. Theoretical Results

Up to now, we have shown how label inference can be reduced to the problem of skill estimation, and addressed the skill estimation problem as a sparse rank-one matrix factorization problem (with noise). In this section, we analyze which properties of the interaction graph ensure learnability as the number of tasks approaches infinity. Subsequently, we analyze the convergence properties of the PGD algorithm for finite tasks.

5.1. Learnability

We start with the analysis for the infinite instance where $C_{ij} = \tilde{C}_{ij} = s_i s_j$ (see Eq. (3) for a definition). There are different ways to let the number of tasks approach infinity while keeping an interaction graph fixed.

Case A: For a fixed interaction graph $G = ([W], E)$ we can consider assignment sets such that the minimum number of shared tasks, $T_{\min}(T) = \min_{(i,j)\in E} N_{ij}(T)$ approaches infinity. Learnability in this context is a property of the interaction graph.

Case B: We can also consider an infinite assignment set A and define $G_A^\infty = ([W], E)$ as the graph where two workers are connected by an edge if $N_{ij} = \infty$. In other words, we define connectivity based on whether two workers interact finitely or infinitely many times.

We will follow the second approach as it is slightly more general than the first (the second approach allows assignment sets A where some workers interact only finitely many times, while the first approach does not allow such assignment sets). Thus, we fix an assignment set A, we let Θ_A be the set of instances sharing assignment set A, and we will consider the learnability of subsets $\Theta \subset \Theta_A$.

To express complete ignorance towards the true unknown labels assigned to tasks, we will consider Θ which are *truth-complete*: informally, this means that Θ places no constraints on what the ground truth could be. Formally, truth completeness means that, for any $\theta = (s, A, g) \in \Theta$, we require $\Theta_{s,A} \subset \Theta$ where $\Theta_{s,A} = \{(s, A, g) \,:\, g \in \{-1, +1\}^{\mathbb{N}}\}$. Truth-completeness expresses that there is no prior information about the unknown labels.

As discussed before, the inference problem is inherently symmetric: the likelihood assigned to some observed data Y under an instance $\theta = (s, A, g)$ is the same as under the instance $(-s, A, -g)$. Thus, an instance set cannot be learnable unless somehow these symmetric solutions are ruled out.

To express the condition this forces us to adopt will require a few more definitions. In particular, given Θ we let $S(\Theta) = \{s \in [-1, 1]^W \,:\, (s, A, g) \in \Theta\}$ be the set of skill vectors that are present in at least one instance in Θ. For a skill vector $s \in [-1, 1]^W$ we let $P(s) = \{i \in [W] \,:\, s_i > 0\}$ be the set of workers whose skills are positive and we let $\mathcal{P}(s) = \{P(s), P(-s)\}$ be the (incomplete) partitioning of workers into workers with positive and negative skills; note that workers with zero skill are left out.

With these definitions in place, we will say that Θ is *rich* if there exists $s \in [-1, 1]^W$ and $\alpha > 1$ such that $\times_{i\in[W]}\{\alpha s_i, s_i/\alpha\} \subset S(\Theta)$ (in other words, there must exist $s \in S(\Theta)$ and $\alpha > 1$ such that we can scale each component of s by either α or $1/\alpha$ and remain in $S(\Theta)$). This is a fairly mild condition; it is satisfied if, for instance, there is some point in $s \in S(\Theta)$ such that a small open-set around that point that is fully contained in Θ.

Richness is required so that there is sufficient ambiguity about skills. Indeed, if richness is not satisfied, then either every skill vector in $S(\Theta)$ is a spammer or hammer (i.e., $s_i \in \{-1, 1\}$ for all i) or $S(\Theta) \cap (-1, 1)^W$ has a specific structure. This structure could potentially be exploited by an algorithm. One might say that assuming richness requires an algorithm to be agnostic to any specific structural knowledge of skill vectors.

We are now ready to state our first main result, which characterizes when rich, truth-complete sets are learnable.

Theorem 8 (Characterization of learnability) *Fix an infinite assignment set A and assume that $G = G_A^\infty$ is connected. Then, a rich, truth-complete set of instances $\Theta \subset \Theta_A$ over A is learnable if and only if the following hold:*

(i) For any $s, s' \in S(\Theta)$ such that $|s| = |s'|$ and $\mathcal{P}(s) = \mathcal{P}(s')$, it follows that $s = s'$;

(ii) The graph G is non-bipartite, i.e, it has an odd-cycle.

Condition (i) requires that any $s \in \Theta$ should be uniquely identified by $|s|$ and knowing which components of s have the same sign and which components are zero. For example, this condition will be met if Θ is restricted so that it only contains skill vectors that have a positive sum. For an explanation of why such an assumption is needed, see the (boldfaced) remark in Section 3.

We remark that if the graph G is not connected, we can simply apply this theorem to each of its connected components. For example, in the situation where none of the workers $1, \ldots, k$ have shared a task with any of the workers $k+1, \ldots, n$, one could try to simply recover the skills of workers $1, \ldots, k$ from their common tasks and then the skills of $k+1, \ldots, n$ from their common tasks. This allows us to drop the condition in the theorem that G be connected, at the expense of changing (ii) to the assertion that none of the connected components of G should be bipartite.

The forward direction of the theorem statement hinges upon the following result which is proved in Appendix:

Lemma 9 *For any $g \in \{\pm 1\}$, $s \in [-1,1]^W$ and an assignment set with a connected, non-bipartite interaction graph G_A^∞, there exists a method to recover $|s|$ and $\mathcal{P}(s)$.*

The reverse implication in the theorem statement follows from the following result:

Lemma 10 *Assume that the lengths of all cycles in G are even. Then there exists $s, s' \in [-1,1]^W$, $s \notin \{-s', s'\}$ such that $C_{ij} = s_i s_j = s'_i s'_j$.*

Learnability for Finite Tasks: We mention in passing that asymptotic learnability is a fundamental requirement, which if not met precludes any reasonable finite time result. In consequence there is no inference schema γ achieves zero regret in this case.

5.2. Convergence of the PGD Algorithm

The previous section established that for learnability the limiting interaction graph G_A^∞ must be a non-bipartite connected graph. We will now show that PGD under these assumptions converges to a unique minimum for both the noisy and noiseless cases.

By the noiseless case, we mean that in the loss L of Theorem 6, we set $\tilde{C}_{ij} = C_{ij} = s_i s_j$ for $(i,j) \in E$. That is, we have infinite number of common tasks to estimate $\tilde{C}_{ij}$ which will then *equal* the expected value C_{ij}. However, in reality, we always suffer from the estimation error (i.e., $|C_{ij} - \tilde{C}_{ij}|, \forall (i,j) \in E$) which leads to a more troublesome problem than a rank one matrix completion. We also provide an analysis of our PGD algorithm in this "noisy" case.

Our first step is to show that, under the condition G is connected and non-bipartite, the loss has a unique minimum and the PGD algorithm recovers the skill vector. For technical convenience, our theorem below considers recovering the absolute values of skills $|s|$. This is the same as recovering the vector s, as we discuss next.

Indeed, observe that if $A = ss^T$ is rank-1, then $|A| = |s||s|^T$ is also rank-1, where the absolute value is taken elementwise. Then we can simply take the absolute value of all-revealed entries; the theorem below will ensure that the PGD method recovers $|s|$. Once $|s|$ is recovered, we need to do some post-processing to recover the sign of each entry.

It should be natural that, because we do not assume access to any true labels, recovering s once $|s|$ is available will require some assumption on the vector s. Indeed, even in the simplest scenario of a complete worker-task interaction graph with $W - 1$ agents always agreeing with each other and disagreeing with agent W, we cannot distinguish between the possibilities that $(s_1 = s_2 = \cdots = s_{W-1} = 1, s_W = -1)$ and $(s_1 = s_2 = \cdots = s_{W-1} = -1, s_W = 1)$. In other words, we fundamentally cannot know if the $W - 1$ agreeing agents are lying or telling the truth. Therefore we will be assuming as before that $\sum_{i=1}^{W} s_i > 0$; this is just saying that there is more truth-telling that lying in the entire system. Under this condition, a post-processing step becomes possible.

Post-processing for sign recovery. If $|s|$ is recovered, we recover the signs of each entry as follows. We assign a positive sign to the first worker ($s_1 > 0$). We then inspect all the elements $s_1 s_j$ over j neighbors of the first worker (i.e., workers that have a joint task with the first worker) and assign a sign to them by inspecting the sign of $s_1 s_j$. We then repeat this, assigning signs to all the neighbors of workers whose sign was just assigned, until the sign of every worker is assigned. Finally, we check if for the resulting vector satisfies $\sum_i s_i > 0$; if not, we flip the sign of every worker. It is immediate that this process always recovers the signs correctly, provided the underlying graph is connected and non-bipartite and $\sum_i s_i > 0$.

Theorem 11 *The PGD Algorithm with $s > 0$ and $x(0) > 0$ converges to the global minimum $x = s$ under conditions for learnability of Theorem 8 and small enough stepsize η.*

As above, if the underlying graph G is not connected, we can, as remarked earlier, simply apply this theorem to each of the connected components. The key requirement of Theorem 8 – that the underlying graph is not bipartite – then becomes the requirement that none of the connected components of G are bipartite.

We next consider the problem of obtaining a polynomial-time convergence rate for the problem, and moreover doing so in the noisy case. Thus we now consider the case when only the perturbed entries $s_i s_j + \Delta_{ij}$ are revealed. We will now make the slightly stronger assumption (discussed at more length below) that the correct answer s lies in the cube $\mathcal{C} = [\kappa, K]^W$ where $0 < \kappa \leq K$. Without loss of generality, we can therefore assume that all revealed entries lie in this set, i.e.,

$$s_i s_j + \Delta_{ij} \in [\kappa, K] \text{ for all } (i, j) \in \Omega,$$

because otherwise we can simply threshold the revealed entries over $[\kappa, K]$ while simultaneously reducing the disturbances Δ_{ij}.

It is natural to attempt to generalize our earlier PGD approach to this setting, in particular by doing gradient descent on the "perturbed" function

$$f_\Delta(x) := \frac{1}{2}\sum_{i,j=1}^{W} N_{ij}(x_i x_j - s_i s_j - \Delta_{ij})^2.$$

While this is possible, we pursue a shortcut naturally adapted to this setting, by using a re-scaling of so-called exponentiated gradient method.

Specifically, defining ∇_t by

$$[\nabla_t]_i = \sum_{j=1}^{W} N_{ij}(x_i x_j - s_i s_j - \Delta_{ij}), \qquad i = 1, \ldots, W,$$

we update as

$$x(t+1) = P_{\mathcal{C}}\left[x(t)e^{-\alpha\nabla_t}\right]. \tag{4}$$

We may think of ∇_t as related to, but not identical, to the gradient of the perturbed function $\nabla f_\Delta(x(t))$. Indeed, observe that the latter quantity will weigh each term in the definition of ∇_t slightly differently. Exponentiated gradient methods of this type are common when optimizing over the simplex, where they come from regularization with the KL divergence (see Hazan (2016)). They are somewhat less common when optimizing over a cube, as we do here. All the same, the following theorem shows that this method is able to achieve polynomial-time convergence for the perturbed problem.

Before we state our main result on the performance of this scheme, we need to introduce some notation. Note that the condition that G is connected and non-bipartite implies that the worker-interaction count matrix N is irreducible and aperiodic. The *signless Laplacian* matrix is then defined as

$$[L_\mathrm{s}]_{ij} = \begin{cases} N_{ij} & j \neq i \\ \sum_{k=1}^{n} N_{ik} & j = i \end{cases}.$$

By contrast, we will use N to denote the matrix whose i, j'th entry is N_{ij}; the matrix N will thus have zero diagonal.

The matrix L_s contrasts with the usual Laplacian because the off-diagonal elements have positive signs. It can be shown that if the graph G is not bipartite, the matrix L_s is positive definite (Desai and Rao, 1994). In fact, the following stronger assertion is true. We will use λ to denote the smallest eigenvalue of the signless Laplacian matrix of a non-bipartite graph with unit weights; we remark that it as consequence of the results of Desai and Rao (1994) that $\lambda \geq 1/W^3$ (where, recall, W is the number of workers, so that the matrices N and L_s are $W \times W$). Finally, we let N_min be the smallest positive weight among $\{N_{ij}\}$. Our final main result, which obtains a polynomial-time convergence rate in both the unperturbed and perturbed cases, is given in the following theorem.

Theorem 12 *Suppose s is located in the interior of $[\kappa, K]^W$ where $0 < \kappa \leq K$. Provided $\max_{i,j} |\Delta_{ij}|$ is small enough and $\alpha = (2\sqrt{W}||N||_2 K^2)^{-1}$, we have that:*

1. *Eq. (4) has a limit, which we will denote by x^*_Δ.*

2. *Convergence to any neighborhood of x_{Δ^*} occurs in polynomial-time.*

3. *x^*_Δ is close to s in the following sense:*

$$||x^*_\Delta - s||_2 \leq K \frac{\sqrt{W}||N||_\infty}{\mu} \max_{i,j} |\Delta_{ij}|,$$

with

$$\mu = \kappa^2 \lambda_{\min}(L_s) N_{\min}.$$

In particular, in the noiseless case when $\Delta = 0$, we have that $x^*_0 = s$.

We note that the assumptions of Theorem 11 are slightly weaker than the assumptions of Theorem 12. While the former assumed that $s > 0$, the latter assumed the slightly stronger statement that $\min_i s_i > \kappa > 0$. This can always be accomplished by throwing out nodes with $s_i \approx 0$ from the data set; such modes are making random guesses and do note contribute to the accuracy of the Bayes classifier of Eq. (1) which should assign them zero weight. A natural way to do this is to simply set to zero any N_{ij} corresponding to correlations $\widetilde{C}_{ij}$ whose absolute values are smaller than $\delta + O(\sqrt{(\log W)/T})$ for some small $\delta > 0$. The advantage of this threshold is that all agents with $s_i = 0$ will, with high probability, have all their interactions N_{ij} set to zero and thus automatically ignored by both the PGD method and the variant of exponentiated gradient proposed here. On the other hand, any pair of workers i, j with $|s_i|$ and $|s_j|$ strictly larger than $\sqrt{\delta}$ will have their correlation above this threshold with high probability .

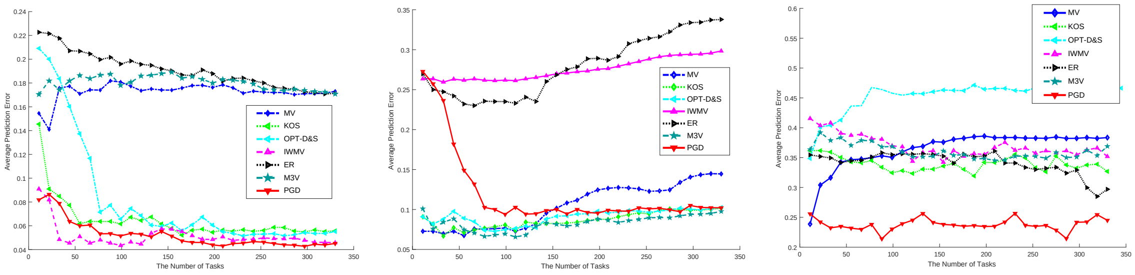
Finally, we discuss the key "trick" underlying the proof of this theorem. The main idea is to interpret the update of Eq. (4) as a projected gradient descent on the function

$$g_\Delta(z) = \frac{1}{2} \sum_{i,j=1}^{W} N_{ij} e^{z_i + z_j} - \sum_{i=1}^{W} z_i \sum_{j=1}^{n} N_{ij}(s_i s_j + \Delta_{ij}),$$

after a change of variable. The construction of this function is what allows us to bypass a lot of the technical difficulties in the analysis.

Finite-Task Bound: Note that we can directly apply this result to obtain a finite task characterization as well. In particular consider a connected and non-bipartite interaction graph. Define $d_{\max}$ as the maximum degree and D as the sum of the degrees. It follows by standard Hoeffding bounds that with probability greater than $(1-\delta)$ we have $\max_{(i,j)\in E} |C_{ij} - \hat{C}_{ij}| \leq \frac{\log(D/\delta)}{\sqrt{N_{\min}}}$. We can then set $\Delta = C_{ij} - \hat{C}_{ij}$ and plug this bound into the above theorem to obtain

$$||x^*_\Delta - s||_2 \leq K \frac{\sqrt{W}||N||_\infty}{\mu} \frac{\log(D/\delta)}{\sqrt{N_{\min}}}.$$



(a) Clique.

(b) Star graph with 3-cycle.

(c) Ring.

Figure 1: Illustrative comparisons of prediction performance for three graph types. Only mean values are plotted for exposition. For the clique, standard deviation values with 11 tasks were 0.09, 0.10, 0.14, 0.14, 0.19, 0.09, and 0.09 for MV, KOS, OPT-D&S, PGD, ER, IWMV, and M3V respectively; and with 330 tasks they were 0.02, 0.01, 0.017, 0.014, 0.012, 0.013, and 0.018 respectively. For the star-graph the standard deviations for 11 tasks were 0.09, 0.13, 0.13, 0.06, 0.13 0.09, and 0.07 for MV, KOS, OPT-D&S, PGD, ER, IWMV, M3V respectively and for 330 tasks they were 0.016, 0.015, 0.013, 0.012, 0.04, 0.03, and 0.013. For the ring the standard deviation for 11 tasks were 0.096, 0.05, 0.08, 0.1, 0.11, 0.09, 0.08 and for 330 tasks they were 0.017, 0.05, 0.02, 0.043, 0.05, 0.086, 0.05. Standard deviations decrease with growing number of tasks.

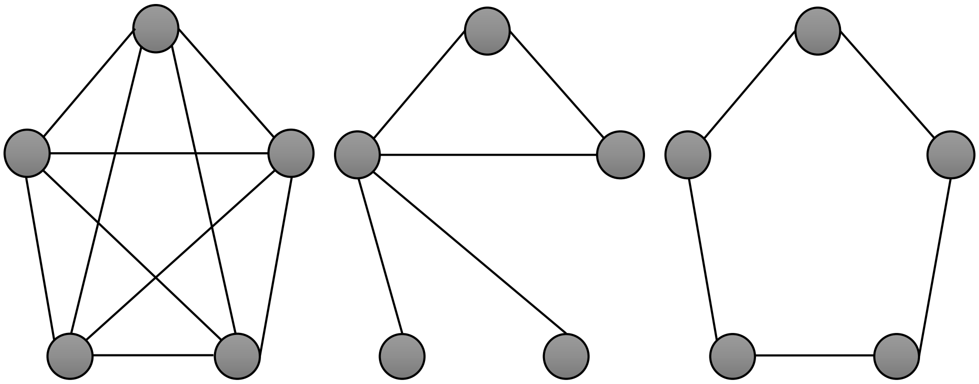
6. Experimental Results

In this section, we will be showing the experimental results to the PGD scheme. We will make one minor modification to the algorithm by adding a projection away from the boundary of the cube $s_i = 1$ by projecting x_i onto $[-1 + t/\sqrt{N_i}, 1 - \tau/\sqrt{N_i}]$ at every step, where recall N_i is the number of tasks assigned to agent i and τ is a parameter. The justification is that skills close to one or negative one have an overwhelming impact on the plug-in rule of Eq. (1). According to Hoeffding inequality, the skill estimates are expected to have an uncertainty proportional to $\tau/\sqrt{N_i}$ with probability const $\times\, e^{-\tau^2}$. There is little loss in accuracy in confining the parameter estimates to the appropriately reduced hypercube, while in principle one could tune this parameter, we use $\tau = 1$ in this paper.

Synthetic Experiments: We will experiment with different graph types, increasing levels of label noise, graph-size, skill distribution, and different weighting functions on synthetic data.

Impact of Graph Type: We consider three 11-node (# workers) irreducible, non-bipartite graphs, namely, a Clique (G_1), Star with augmented odd cycle (G_2), and a Ring (G_3) to illustrate the impact of sparsity (Clique has dense worker interactions while Star/Ring have fewer than 3 worker interactions) and graph-type (Ring vs. Star). An illustration of the different graph types is shown in Figure 2. These graphs satisfy condition (ii) of Thm 8.

Noise Robustness: To see the impact of noise, we vary the noise level by increasing the number of tasks, which in turn reduces the error in the correlation matrix. Tasks are



(a) Clique. (b) Star graph with 3-cycle. (c) Ring.

Figure 2: Different Graph Types.

Type of workers	α	β	Bayes error	Prediction error (const. noise)
Adversary vs. hammer	0.5	0.5	0.0036 ± 0.0014	0.5990 ± 0.4860
Asym. with more positive skills	5	1	0.0038 ± 0.0014	0.0041 ± 0.0013
Asym. with more negative skills	2	5	0.0314 ± 0.0062	0.9667 ± 0.0067
Hammer	2	2	0.0615 ± 0.0083	0.4162 ± 0.4273
Spammer	1	3	0.0129 ± 0.0034	0.9864 ± 0.0041

Table 1: Average prediction errors with different skills distributions.

randomly assigned to binary classes ± 1 with total number of tasks ranging from 11 to 330. Skills are randomly assigned on a uniform grid between 0.8 and -0.3[2]

Influence of Different Skill-Distribution: We randomly assign binary classes to $T = 300$ tasks and select five pairs of parameters. Average prediction errors are presented in Table 1 averaged over 10 independent runs. Parameters $\alpha = 5, \beta = 1$, correspond to reliable workers leading to small prediction error; the prediction error with parameters $\alpha = 2, \beta = 2$ and $\alpha = 0.5, \alpha = 0.5$, is almost random because of $\sum_{i \in [W]} s_i$ is no longer positive, which validates our theory. Similar situation arises for $\alpha = 2, \beta = 5$ and $\alpha = 5, \beta = 1$, because the skills are all flipped relative to our assumption that the sum of the skills is positive.

Influence of Graph Size: We focus on how the graph size affects the performance of PGD algorithm. Note that graph size is associated with the number of workers. Our goal is to demonstrate that for a constant amount of noise, prediction accuracy of PGD does not degrade with graph-size. We again consider the case when the worker-interaction graph is a star-graph with an odd-cycle of length 3. We increase the size of worker-interaction graph by adding nodes to the star-graph. Skills s are selected between 0.8 and -0.3 uniformly. To fix the noise level, we define $C_{ij} = s_i s_j + \xi_{ij}, \forall (i,j) \in E$ where ξ_{ij} is randomly selected from $[-0.2, 0.2]$. Note that the noise level is quite large relative to what we expect in terms of accuracy of correlation estimates. We iteratively run PGD for 50 times. The average

[2] The reason for this choice is to satisfy condition (i) in Theorem 8, i.e., requiring overall skills to be positive. Aggregate skill is about 0.25.

Number of workers	21	51	71	91
Bayes error	0.0425 ± 0.0042	0.0622 ± 0.0040	0.0634 ± 0.0033	0.0574 ± 0.0030
Prediction error (const. noise)	0.0425 ± 0.0042	0.0641 ± 0.0126	0.0662 ± 0.0072	0.0618 ± 0.0063

Table 2: Average prediction errors for different graph sizes.

Table 3: Prediction errors for different weightings

Worker type Assigned most tasks	$[N_{ij} > 0]$	$B(N_{ij}) = N_{ij}$	$B(N_{ij}) = N_{ij}^2$
Spammers	0.33 ± 0.03	0.33 ± 0.03	0.55 ± 0.17
Positive skill workers	0.17 ± 0.06	0.09 ± 0.02	0.09 ± 0.02

prediction errors with different graph size it presented in Table 2. It can be seen that the prediction error is not sensitive to the graph size compared to the Bayes error.

Influence of Weighting function: It is straighforward from our proof of Theorem 11 to see that PGD algorithm converges to the global optimal for any non-negative weights. Our objective is based on weighting with number of counts in Eq. 6. However, there are other options that one could consider. Dalvi et al. (2013) has suggested using $B(N_{ij}) = N_{ij}^2$, while we use N_{ij}. Another possibility is to use binary weights. We iteratively run PGD 10 times for each weighing function with $T = 300$ tasks for different types of task assignements. If N_{ij}'s are all equal, these choices produce identical results. We consider two cases: (a) Spammers are assigned a majority of tasks; (b) Positively skilled workers are assigned most tasks. The prediction errors are compared in Table 3. Note that quadratic weighting is quite bad in this case because it tends to ignore positively skilled workers. On the other hand unweighted case does not accurately estimate spammers and also results in poor choice.

We compare the average prediction error $PE = \frac{1}{T} \sum_{t=1,\ldots,T} 1\{\hat{Y}_t \neq g_t\}$ with the Majority Voting (MV) algorithm, the KOS algorithm (Karger et al., 2013), Opt-D&S algorithm (Zhang et al., 2014), the ER algorithm (Dalvi et al., 2013), the IWMV algorithm (Li and Yu, 2014), and the M3V algorithm (Tian and Zhu, 2015). The KOS algorithm is based on belief propagation, Opt-D&S uses a spectral method to initialize EM, the ER algorithm is the more successful spectral method of the paper defining it, the IWMV algorithm is an EM-style algorithm. Each algorithm is averaged over 15 trials on each dataset. The average prediction errors are presented in Figure 2. As the number of tasks grows, the average prediction error of PGD algorithm decreases. PGD is evidently robust to missing data/sparsity and graph-type. OPT-DS, which is close to PGD performance suffers significant performance degradation on sparse graphs such as rings. We can attribute this to the fact that a tensor-based method requires at least 3 worker annotations for each task Zhang et al. (2014).

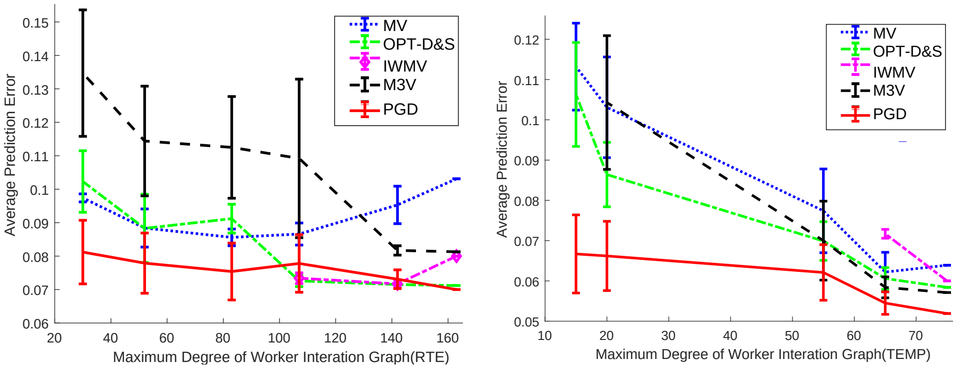
Benchmark Dataset Experiments: We illustrate the performance of PGD algorithm against state-of-art algorithms. Each algorithm is executed on five data-sets, i.e. RTE1 (Snow et al.), Temp (Snow et al.), Dogs (Deng et al., 2009), WSD (Word Sense Disambiguation) (Snow et al.),and WebSearch (Zhou et al., 2012). A summary of these data-sets is presented in Table 4. Following convention we report errors between ground-truth and recovered labels in Table 5. Note that on WSD dataset, OPT-D&S algorithm does not converge to a equilibrium point after 1000 iterations.

Table 4: Summary of Benchmark Datasets.

Datasets	Tasks	Workers	Instances	Classes	Sparsity level
RTE1	800	164	8000	2	0.0610
Temp	462	76	4620	2	0.1316
Dogs	807	109	8070	4	0.0917
WSD	177	34	1770	3	0.2947
Web	2665	177	15567	5	0.0033

Table 5: Prediction Errors of Different Methods.

Data	PGD	MV	Opt-D&S	KOS	ER	IWMV	M3W
RTE1	**0.07**	0.1031	0.0712	0.3975	0.14	0.08	0.0813
Temp	**0.0512**	0.0639	0.0584	0.0628	0.052	0.06	0.0606
Dogs	**0.1660**	0.1958	0.1689	0.3172	0.18	0.19	0.1822
WSD	0.0056	0.0056	N/A	0.0056	0.0056	0.0056	0.0056
Web	**0.1485**	0.2693	0.1586	0.4293	0.22	0.22	0.1847



(a) RTE1.

(b) TEMP.

Figure 3: Impact of Graph Sparsification.

Influence of Graph Sparsification: Here we consider the scenario where fewer workers label each task on the binary classification benchmark datasets. Binary classification tasks are aligned with our theoretical results. This experiment will highlight the performance of state-of-art algorithms under sparse task-assignments. We simulate this effect based on random sparsification. In particular, we sort the degree of each node on the interaction graph. To sparsify the graph we randomly delete edges starting with the highest degree node and continue this process for other nodes until we obtain an interaction graph with desired maximum degree. We also remove symmetrically remove corresponding edges of incident workers to maintain symmetry. This has the implicit effect of deleting some of the tasks as well (for instance, if a task is annotated by two workers). Higher levels of sparsification leads to fewer availability of tasks for training. We iteratively run PGD and the other algorithms for 50 Monte-Carlo trials with different desired maximum degrees. The average prediction errors are displayed in Figure 3. The reason IWMV performs poorly is that majority votes are no longer reliable, which IWMV relies on. Our PGD algorithm is surprisingly robust to sparsification of interactions and degrades gracefully relative to other schemes. This highlights the fact that PGD is capable of leveraging sparse interactions among workers and obtain fairly robust estimates of skill-levels required for accurate prediction.

Normal gradient descent vs Eq. (4): Although we have found it convenient to use Eq. (4) for our polynomial-time convergence results, in practice we do not see much advantage of that iteration compared to normal gradient descent. Figure 4 gives a comparison in the noiseless case while Figure 5 gives a comparison in the noisy case. The results are extremely similar. Results are shown for random s and three choice of graphs; each plot is the result of a single realization. In general, all the realizations we have seen look like the plots shown, with only minor differences between the methods.

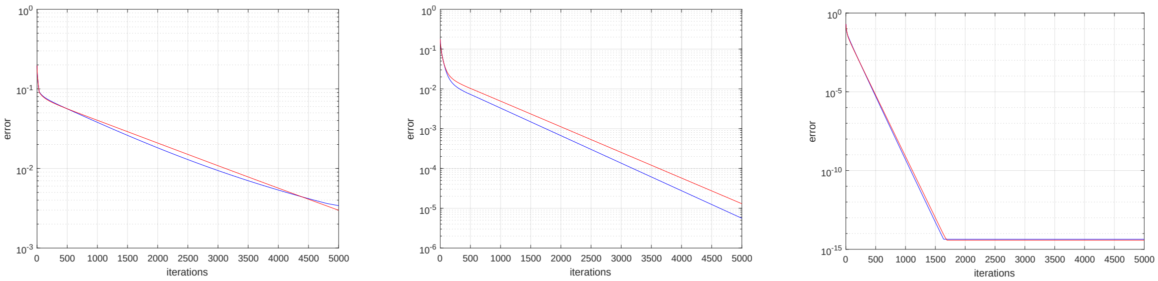


Figure 4: **Noiseless case:** the figures show gradient descent in blue vs the variant of exponentiated gradient we have proposed in Eq. (4) in red. The plots show the error $||x(t) - s||_\infty$ on the y-axis vs the iteration t on the x-axis. The circle graph is shown on the left, the 2D grid with one extra edge in the middle, and an Erdos-Renyi random graph on the right. On all cases, the number of workers is 25. A single edge was added to the 2D grid in order to ensure aperiodicity. All the perturbations are zero and the correct answer s was generated with each component uniformly random over $[0.4, 0.6]$. Starting point was $x(0) = 0.6 \cdot \mathbf{1}$ and stepsize was taken to be 0.01 in both cases.

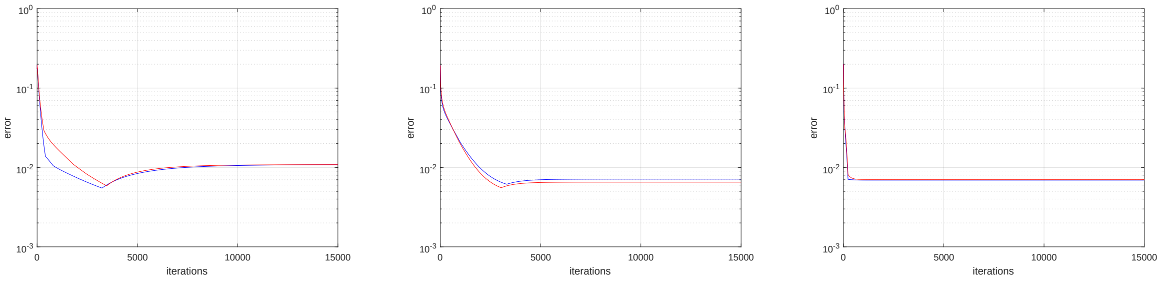


Figure 5: **Noisy case:** Everything is the same as in Figure 4, except that small independent noises of $0.01X$ where X is uniform on $[-1/2, 1/2]$ is added to every entry. This results in a perturbation matrix Δ with $||\Delta||_2 \sim 10^{-2}$, and all the final deviations from the correct solution s are also on the order of 10^{-2}.

Table 6: Time Complexity/Iteration of Different Methods.

Alg.	PGD	IWMV	M3W
Com.	$O(D_{max}W)$	$O(TW)$	$O(W^2T)$

Time Complexity: We also compare the time complexity of proposed algorithm against state-of-art algorithms. Our PGD algorithm requires fewer iterations in comparison to other iterative methods and each iteration scales linearly with W and the maximum degree, D_{max}, of the worker-interaction graph which is bounded by W. Time complexity of different algorithms is summarized in Table 6 [3].

7. Conclusions

We propose a new moment-matching approach with weighted rank-one approximation and propose a gradient algorithm for worker skill estimation in Crowdsourcing. In contrast to prior work, the weights are set up to correct for the spread of the measured worker-worker agreements accuracies which are typical in real-world problems where who works on the same task with whom is out of control. Our results explicitly characterize identifiability and convergence rates in terms of spectral graph theoretical quantities, revealing the importance of worker interaction graphs for skill estimation. The general problem studied here, is related to state estimation with intermittent and active sensor communications (Saligrama and Castanon, 2006; Hanawal et al., 2017), which we plan to explore in future work.

References

P. S. Albert and L. E. Dodd. A cautionary note on the robustness of latent class models for estimating diagnostic error without a gold standard. *Biometrics*, 60(2):427–435, 2004.

[3] Opt-D&S, KOS, and ER algorithms are omitted. They employ spectral factorization and have high time complexity.

D. Berend and A. Kontorovich. Consistency of weighted majority votes. In *NIPS*, pages 3446–3454, 2014.

T. Bonald and R. Combes. Crowdsourcing: Low complexity, minimax optimal algorithms. arXiv preprint arXiv 1606.00226, 2016.

N. Dalvi, A. Dasgupta, R. Kumar, and V. Rastogi. Aggregating crowdsourced binary ratings. In *WWW*, pages 285–294, 2013.

A. P. Dawid and A. M. Skene. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Applied Statistics*, 28(1):20–28, 1979.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

M. Desai and V. Rao. A characterization of the smallest eigenvalue of a graph. *Journal of Graph Theory*, 18(2):181–194, 1994. doi: 10.1002/jgt.3190180210.

R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*, volume xx. 01 2001. ISBN 0-471-05669-3.

K. R. Gabriel and S. Zamir. Lower rank approximation of matrices by least squares with any choice of weights. *Technometrics*, 21(4):489–498, November 1979. ISSN 0040-1706. doi: 10.1080/00401706.1979.10489819.

C. Gao and D. Zhou. Minimax optimal convergence rates for estimating ground truth from crowdsourced labels. arXiv preprint arXiv:1310.5764, 2013.

C. Gao, Y. Lu, and D. Zhou. Exact exponent in optimal rates for crowdsourcing. In *ICML*, pages 603–611, 2016.

R. Ge, J. Lee, and T. Ma. Matrix completion has no spurious local minimum. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 2973–2981. Curran Associates, Inc., 2016.

A. Ghosh, S. Kale, and P. McAfee. Who moderates the moderators? crowdsourcing abuse detection in user-generated content. In *Proc. of the 12th ACM conference on Electronic commerce*, pages 167–176, 2011.

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.

M. Hanawal, C. Szepesvári, and V. Saligrama. Unsupervised sequential sensor acquisition. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA*, pages 803–811, 2017.

Elad Hazan. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016.

S. L. Hui and S. D Walter. Estimating the error rates of diagnostic tests. *Biometrics*, pages 167–171, 1980.

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

D.R. Karger, S. Oh, and D. Shah. Efficient crowdsourcing for multi-class labeling. In *SIGMETRICS*, pages 81–92, 2013.

D.R. Karger, S. Oh, and D. Shah. Budget-optimal task allocation for reliable crowdsourcing systems. *Operations Research*, 62(1):1–24, 2014.

Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 2009.

H. Li and B. Yu. Error rate bounds and iterative weighted majority voting for crowdsourcing. 11 2014.

Q. Liu, J. Peng, and A. T. Ihler. Variational inference for crowdsourcing. In *NIPS*, pages 692–700, 2012.

Y. Nesterov. *Introductory Lectures on Convex Optimization*. Kluwer Academic Publishers, 2004.

S. Nitzan and J. Paroush. The characterization of decisive weighted majority rules. *Economics Letters*, 7(2):119–124, 1981.

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

V. Saligrama and D. A. Castanon. Reliable distributed estimation with intermittent communications. In *Proceedings of the 45th IEEE Conference on Decision and Control*, Dec 2006.

L. Shapley and B. Grofman. Optimizing group judgmental accuracy in the presence of interdependencies. *Public Choice*, 43(3):329–343, January 1984. ISSN 0048-5829, 1573-7101. doi: 10.1007/BF00118940.

P. Smyth, U. Fayyad, M. Burl, P. Perona, and P. Baldi. Inferring ground truth from subjective labelling of venus images. In *NIPS*, volume 7, pages 1085–1092, 1995.

R. Snow, B. O'Connor, D. Jurafsky, and A. Ng. Cheap and fast—but is it good?: Evaluating non-expert annotations for natural language tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 254–263, Stroudsburg, PA, USA. Association for Computational Linguistics.

N. Srebro and T. Jaakkola. Weighted low-rank approximations. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 720–727, 2003.

D. Szepesvári. A statistical analysis of the aggregation of crowdsourced labels. Master's thesis, University of Waterloo, 2015.

T. Tian and J. Zhu. Max-margin majority voting for learning from crowds. In *Advances in Neural Information Processing Systems 28*, pages 1621–1629. Curran Associates, Inc., 2015.

Y. Zhang, X. Chen, D. Zhou, and M.I. Jordan. Spectral methods meet EM: A provably optimal algorithm for crowdsourcing. In *NIPS*, pages 1260–1268, 2014.

Y. Zhang, Xi Chen, Dengyong Zhou, and Michael I. Jordan. Spectral methods meet EM: A provably optimal algorithm for crowdsourcing. *The Journal of Machine Learning Research*, 17(1):3537–3580, 2016.

D. Zhou, S. Basu, Y. Mao, and J.C. Platt. Learning from the wisdom of crowds by minimax entropy. In *NIPS*, pages 2195–2203, 2012.

Appendix

Appendix A. Proof of Lemma 6

Recall that θ is a finite index. For each (i,j) such that $N_{ij} > 0$, we have

$$\begin{aligned}\frac{1}{N_{ij}} \sum_{t:(i,t),(j,t)\in A} (Y_{i,t}Y_{j,t} - x_i x_j)^2 &= \frac{1}{N_{ij}} \sum_{t:(i,t),(j,t)\in A} (1 - 2Y_{i,t}Y_{j,t}x_i x_j + x_i^2 x_j^2) \\ &= 1 - 2\tilde{C}_{ij} x_i x_j + x_i^2 x_j^2 \\ &= \tilde{C}_{ij}^2 - 2\tilde{C}_{ij} x_i x_j + x_i^2 x_j^2 + 1 - \tilde{C}_{ij}^2 \\ &= (\tilde{C}_{ij} - x_i x_j)^2 + 1 - \tilde{C}_{ij}^2.\end{aligned}$$

Therefore,

$$\frac{1}{2} \sum_{(i,t),(j,t)\in A} (Y_{i,t}Y_{j,t} - x_i x_j)^2 = \frac{1}{2} \sum_{i,j\in[W]} N_{ij}(\tilde{C}_{ij} - x_i x_j)^2 + \sum_{i,j\in[W]} N_{ij}(1 - \tilde{C}_{ij}^2)$$

Since $\sum_{i,j\in[W]} N_{ij}(1 - \tilde{C}_{ij}^2)$ is a constant, Eq.(2) is equivalent to the optimization problem $\operatorname{argmin}_{x\in[-1,+1]^W} L(x)$.

Appendix B. Proof of Theorem 8

The proof directly follows from Lemma 9 and Lemma 10. We will next prove these Lemmas.

Proof of Lemma 9: Take any two workers i, j that are connected in G. Let $t \in \mathcal{N}$ be a task such that $(i,t),(j,t) \in A$. By assumption,

$$Y_{i,t}Y_{j,t} = g_t^2 Z_{i,t} Z_{j,t} = Z_{i,t} Z_{j,t}.$$

Let us define

$$\bar{C}_{ij} \doteq \lim_{T\to\infty} \frac{1}{T} \sum_{(i,t),(j,t)\in A, t\leq T} Y_{i,t}Y_{j,t}.$$

By the law of large numbers,

$$\bar{C}_{ij} = E[Z_{i,t}Z_{j,t}] = s_i s_j.$$

For convenience, we define $\bar{C}_{ij} = 0$ when $(i,j) \notin E$.

Next without loss of generality assume that workers $1, 2, \ldots, 2k+1$ form a cycle in G. Then,

$$\begin{aligned}s_1 &= \bar{C}_{1,2k+1} s_{2k+1}^{-1} \\ &= C_{1,2k+1} C_{2k+1,2k}^{-1} s_{2k} \\ &= C_{1,2k+1} C_{2k+1,2k}^{-1} C_{2k,2k-1} s_{2k-1}^{-1} \\ &\vdots \\ &= C_{1,2k+1} C_{2k+1,2k}^{-1} C_{2k,2k-1} \ldots C_{2,1} s_1^{-1},\end{aligned}$$

which implies that

$$|s_1| = \sqrt{C_{1,2k+1}C^{-1}_{2k+1,2k}C_{2k,2k-1}\dots C_{2,1}}\,,$$

assuming that $C_{2,3}, C_{4,5}, \dots, C_{2k,2k+1} \neq 0$. This gives a method to recover $|s_1|$.

Now, since G is connected, for any worker i there exists a path from worker 1 to worker i. If this path was given by the vertices $1, 2, \dots, \ell$ then

$$\begin{aligned} |s_\ell| =& |C_{\ell,\ell-1}|\,|s^{-1}_{\ell-1}| = |C_{\ell,\ell-1}|\,|C^{-1}_{\ell-1,\ell-2}|\,|s_{\ell-2}| \\ =& \dots = |C_{\ell,\ell-1}|\,|C^{-1}_{\ell-1,\ell-2}| \cdots |C^{(-1)^\ell}_{2,1}|\,|s_1|^{(-1)^{\ell+1}}\,. \end{aligned}$$

which shows how $|s_l|$ may be recovered. We conclude that $|s|$ can be recovered.

It remains to show that $\mathcal{P}(s)$ can be recovered. Let $i, j \in [W]$ be two different workers. Then, if $\pi \subset E$ is any path in G from i to j, we have

$$\Pi_{(u,v)\in E}\operatorname{sgn}(C_{u,v}) = \Pi_{(u,v)\in E}\operatorname{sgn}(s_u)\operatorname{sgn}(s_v) = \operatorname{sgn}(s_i)\operatorname{sgn}(s_j).$$

We emphasize this holds for all paths connecting i and j, and in particular $\Pi_{(u,v)\in E}\operatorname{sgn}(C_{u,v})$ is the same for any path connecting i and j.

Now if i and j are such that for some path π connecting them $\Pi_{(u,v)\in E}\operatorname{sgn}(C_{u,v}) = +1$, we assign i, j to the same group; otherwise we assign them to different groups. It is easy to see that this creates exactly two groups. The resulting "partition" must match $\mathcal{P}(s)$.

Proof of Lemma 10: Take s, α which are used in the definition of richness of Θ. We construct two other skill vectors s' and s'' as follows: We set $s'_1 = \alpha s_1$ and $s''_1 = s_1/\alpha$. Now, if worker i is at an even distance from worker 1 on some path in G then $s'_i = \alpha s_i$ and $s''_i = s_i/\alpha$, otherwise we set $s'_i = s_i/\alpha$ and $s''_i = \alpha s_i$. Note that all workers can be accessed from worker 1 because G is connected. Note that if there are multiple paths from worker 1 to some other worker then all of these have the same parity, or the graph had an odd cycle. Now, both s and s' give rise to the same products, $s_i s_j$, along any edge $(i, j) \in E$. Since both are in Θ by assumption, the result is proven.

Reverse Direction for Theorem 1: We prove this by contraposition. First, assume that (i) does not hold. We want to prove that learnability fails. If (i) does not hold, we can take $s, s' \in [-1, 1]^W$ different skill vectors such that $|s| = |s'|$ and $\mathcal{P}(s) = \mathcal{P}(s')$ and $s, s' \in S(\Theta)$. It follows that $s = -s'$. Take any $g \in \{\pm 1\}^W$. Note that the instances (s, A, g) and $(-s, A, -g)$ lead to the same joint distribution over the observed labels. Hence, no inference schema can tell these instances apart, thus any inference schema will suffer linear regret on one of these instances. Now, if (ii) does not hold, Lemma 10 gives two skill vectors s, s' which are different and $s \neq \pm s'$, which again give the same likelihood to any data. This again leads to that any inference schema will suffer a linear regret on one of these instances.

Appendix C. Proof of Theorem 11

Our first step is to argue that, with sufficiently small step-size, the PGD method remains bounded. To that end, we have the following proposition.

Proposition 13 *Let*

$$V(x) = \max_{i=1,\ldots,W} \max\left(\frac{x_i}{s_i}, \frac{s_i}{x_i}\right)$$

and suppose that x^t is positive and s is positive and that the positive step-size γ is small enough so that

$$\eta||ND_s^2||_\infty V(x^t)^2 \leq 1.$$

Then, it holds that $V(x^{t+1}) \leq V(x^t)$.

Proof Let us use the notation $a./b$ for the elementwise ratio of vectors a and b, and define $r^t = x^t./s$. As a consequence, $V(x^t) = \max_{i=1,\ldots,n} \max\left(r_i^t, 1/r_i^t\right)$. Now suppose $V(x^t) = Z$, which is a positive number due to the assumed positivity of x^t and s; this implies that

$$Z^{-1} \leq r_i^t \leq Z \text{ for all } i = 1, \ldots, n. \tag{5}$$

Let us fix some index j. The prove the lemma we just need to prove that Eq. (5) holds for r_j^{t+1}.

Suppose first that $r_j^t = \beta Z$ where $\beta \in (0, 1]$. Then

$$\begin{aligned} x_j^{t+1} &= x_j^t - \eta \sum_{k=1}^n N_{jk} x_k^t (x_k^t x_j^t - s_k s_j) \\ &= x_j^t - \eta \sum_{k=1}^n N_{jk} s_k^2 s_j r_k^t (r_k^t r_j^t - 1) \end{aligned}$$

and therefore

$$r_j^{t+1} = r_j^t - \eta \sum_{k=1}^n N_{jk} s_k^2 r_k^t (r_k^t r_j^t - 1) \tag{6}$$

Now since $r_j^t = \beta Z$ and $r_k^t \geq Z^{-1}$ for all k, we have

$$\begin{aligned} r_j^{t+1} &\leq \beta Z - \eta \sum_{k=1}^n N_{jk} s_k^2 r_k^t (Z^{-1} \beta Z - 1) \\ &= \beta Z + \eta(1-\beta) \sum_{k=1}^n N_{jk} s_k^2 r_k^t \\ &\leq \beta Z + \eta(1-\beta)||ND_s^2||_\infty Z \\ &\leq \beta Z + (1-\beta) Z \\ &= Z, \end{aligned}$$

where we used our step-size bound as well as the fact that $V(x^t) \geq 1$ due to the definition of $V(\cdot)$. This proves the upper bound we seek.

For the other direction, suppose $r_j^t = \mu Z^{-1}$ where now $\mu \in [1, \infty)$. From Eq. (6), and using the fact that $r_k \leq Z$ for all k, we then have

$$\begin{aligned} r_j^{t+1} &\geq \mu Z^{-1} - \eta \sum_{k=1}^{n} N_{jk} s_k^2 r_k (Z\mu Z^{-1} - 1) \\ &= \mu Z^{-1} - \eta(\mu - 1) \sum_{k=1}^{n} N_{jk} s_k^2 r_k \\ &\geq \mu Z^{-1} - \eta(\mu - 1)||ND_s^2||_\infty Z \\ &\geq \mu Z^{-1} - \eta(\mu - 1)Z^{-1}||ND_s^2||_\infty Z^2 \\ &\geq \mu Z^{-1} - (\mu - 1)Z^{-1} \\ &= Z^{-1} \end{aligned}$$

■

As a consequence of this proposition, we have the following bound on how big the iterates x^t can get.

Corollary 14 *Suppose s and x^0 belong to $[\kappa, K]^W$ where $0 < \kappa \leq K$. If the step-size η of PGD algorithm satisfies*

$$0 \leq \eta \leq \frac{\kappa^2}{K^2||ND_s^2||_\infty},$$

then $V(x^t) \leq K/\kappa$ and $x^t \in [\kappa^2/K, K^2/\kappa]^W$ for all $t = 1, 2, \ldots$.

Proof Clearly, $V(x^0)^2 \leq (K/\kappa)^2$ given the fact that both s and x^0 belong to $[\kappa, K]$. Then, η satisfies the step-size condition of Proposition 13 at time 0. Using Proposition 13, we can conclude that $V(x^1) \leq V(x^0)$ which implies $V(x^1)^2 \leq V(x^0)^2 \leq (K/\kappa)^2$ as well. By applying the same technique iteratively, we have $V(x^t) \leq \ldots \leq V(x^1) \leq V(x^0) \leq K/\kappa, \forall t$. Since $s \in [\kappa, K]^W$, this implies $x^t \in [\kappa^2/K, K^2/\kappa]^W$. ■

A consequence of the last corollary result is that the function

$$L_{\text{noiseless}}(x) \doteq \frac{1}{2} \sum_{(i,j)\in E} N_{ij}(s_i s_j - x_i x_j)^2.$$

can, for all practical purposes, be assumed to have a gradient which is Lipschitz. Of course, this is false over all of $\mathbb{R}^n$, but since the iterates of the PGD method stay within a compact set, the gradient of $L_{\text{noiseless}}$ will be Lipschitz over the region of interest. In particular, we have the following estimate.

Proposition 15 *For any $y, z \in [0, K^2/\kappa]^W$, we have that*

$$||\nabla L_{\text{noiseless}}(y) - \nabla L_{\text{noiseless}}(z)||_2 \leq \hat{L}||y - z||_2,$$

with

$$\hat{L} = 4W||N||_F \frac{K^4}{\kappa^2}.$$

Proof We begin by establishing the following claim. **Claim:** Suppose y, z are vectors belonging to the cube $y, z \in [0, A]^W$. Then

$$||N \circ (yy^T - zz^T)||_F \leq 2\sqrt{W} A ||N||_F ||y - z||_2.$$

This proposition could be obtained by a simple algebraic manipulation as

$$\begin{aligned}
||N \circ (yy^T - zz^T)||_F &= \mathrm{Tr}(\mathrm{abs}(N)\mathrm{abs}(yy^T - zz^T)) && (7)\\
&\leq ||N||_F ||yy^T - zz^T||_F && (8)\\
&= ||N||_F ||yy^T - yz^T + yz^T - zz^T||_F && (9)\\
&\leq ||N||_F \left(||y(y - z)^T||_F + ||(y - z)z^T||_F \right) && (10)\\
&\leq 2||N||_F \sqrt{W} A ||y - z||_2 && (11)
\end{aligned}$$

Eq. (7) follows via the inequality $||A \circ B||_F \leq \mathrm{Tr}(\mathrm{abs}(A)\mathrm{abs}(B)^T)$. Eq. (8) is obtained by the Cauchy-Schwarz inequality in the form of $\mathrm{Tr}(AB) \leq ||A||_F |||B||_F$. Eq. (9) and Eq. (10) are self-explanatory. Eq. (11) follows because every entry of the vectors y, z is at most A. This concludes the proof of the claim.

Let $P_x = N \circ (xx^T - ss^T)$. Then $\nabla L(x) = P_x x$ and therefore

$$\begin{aligned}
||\nabla L_{\text{noiseless}}(y) - \nabla L_{\text{noiseless}}(z)||_2 &= ||P_y y - P_z z||_2\\
&= ||P_y y - P_y z + P_y z - P_z z||_2\\
&\leq ||P_y||_2 ||y - z||_2 + ||P_y - P_z|| ||z||_2\\
&\leq 2\sqrt{W} \frac{K^2}{\kappa} ||N||_F ||y - z||_2 ||y - z||_2\\
&\quad +2\sqrt{W} \frac{K^2}{\kappa} ||N||_F ||y - z||_2 ||z||_2,
\end{aligned}$$

where the last step used that the 2-norm of a matrix is upper bounded by its Frobenius norm as well as the above claim. Now using the bound $||z||_2 \leq (K^2/\kappa)\sqrt{W}$ and the same for $||y - z||_2$, we obtain the lemma.

■

We now use the standard fact that, for a function $f(x)$ with L-Lipschitz gradient, gradient descent with step-size $h < 2/L$ generates a sequence which satisfies

$$f(x^{t+1}) = f(x^t) - h \left(1 - \frac{L}{2} h \right) ||\nabla f(x^t)||_2^2.$$

In particular, $\nabla f(x^t) \to 0$ under these conditions if $f(x^t)$ is bounded below.

Clearly, $L_{\text{noiseless}}$ is bounded below by zero. Thus, to argue that gradient descent on $L_{\text{noiseless}}$ results in $x^t \to s$, we just need to argue that the only point that satisfies $\nabla L_{\text{noiseless}}(x) = 0$ is $x = s$.

We complete the proof of Theorem 11 by now proving that last statement.

Observe that

$$[\nabla L_{\text{noiseless}}(x)]_i = \sum_{j=1}^{W} N_{ij}(x_i x_j - s_i s_j) x_j,$$

where the interaction matrix N which is nonnegative, irreducible, symmetric, and with zero diagonal. What we need to argue is that, given $s > 0$, there does not exist $x > 0, x \neq s$ such that for each $i = 1, \ldots, W$, we have

$$\sum_{j=1}^{W} N_{ij}(x_i x_j - s_i s_j) x_j = 0. \tag{12}$$

We begin by adopting the following notation. For a vector x, D_x will refer to the diagonal matrix with x on the diagonal. For a matrix A, $\operatorname{diag}[A]$ will refer to the *diagonal of A stacked as a vector* (note that this is an **unusual** notation). Also, let us refer to the set of matrices which are nonnegative, irreducible, aperiodic, symmetric and with zero diagonal as *admissible*.

Assume that x satisfies Eq. (12). Then, we can multiply the ith equation of (12) by x_i. Our first observation is that we may rewrite Eq. (12) as

$$\operatorname{diag}\Big[D_x N D_x (xx^T - ss^T)\Big] = 0. \tag{13}$$

It suffices to argue that we cannot positive x and admissible F such that

$$\operatorname{diag}\Big[F(xx^T - ss^T)\Big] = 0.$$

Note that we were able to drop the D_xs from the equation because N is admissible if and only if $D_x N D_x$ is.

We proceed as follows. Since

$$x_i x_j - s_i s_j = s_i \left(\frac{x_i}{s_i} \frac{x_j}{s_j} - 1 \right) s_j \,,$$

defining $u_i = x_i / s_i$, we have that u is positive and that

$$xx^T - ss^T = D_s (uu^T - 11^T) D_s \,.$$

We must therefore argue that it is impossible to find $u > 0, u \neq 1$ and admissible F such that

$$\operatorname{diag}\Big[D_s F D_s (uu^T - 11^T) D_s\Big] = 0.$$

Since $s > 0$ it will suffice to argue that we cannot find $u > 0, u \neq 1$ and admissible Z such that

$$\operatorname{diag}\Big[Z(uu^T - 11^T)\Big] = 0. \tag{14}$$

Without loss of generality, we can assume that $u_1 \leq u_2 \leq \cdots \leq u_W$; we can always relabel indices to make this hold.

Now there are three possibilities:

1. $u_1 u_W > 1$.

2. $u_1 u_W = 1$.

3. $u_1 u_W < 1$.

We argue that in each case we cannot find a suitable u that satisfies Eq. (14). Indeed, let us consider the first possibility. In that case the last column of $uu^T - 11^T$, with entries $u_i u_W - 1$, is strictly positive, and therefore, considering that $[Z(uu^T - 11^T)]_{WW} = 0$, we obtain that the last row of Z must be zero – contradicting irreducibility. Similarly, in case 3, the first column of $uu^T - 11^T$, with entries $u_1 u_i - 1$, is negative, and, considering that $[Z(uu^T - 11^T)]_{11} = 0$, we see that the first row of Z must be zero, which can not hold true.

It remains to consider case 2. We may assume that $u_1 < u_W$ (ruling out the possibility that a u proportional to the all-ones vector satisfies Eq. (14) is trivial). We break up $\{1, \ldots, W\}$ into three blocks. The first block is all the indices j such that $u_j = u_1$. The third block is all the indices j such that that $u_j = u_W$. All the other indices go into block 2. Note that block 2 may be empty, for example if every entry of u is equal to u_1 or u_W.

The advantage of partitioning this way is that the matrix $uu^T - 11^T$ has the following sign structure:

$$uu^T - 11^T = \left(\begin{array}{c|c|c} - & - & 0 \\ \hline - & * & + \\ \hline 0 & + & + \end{array}\right)$$

where $-$ represents a strictly negative submatrix, $+$ represents a strictly positive submatrix, while $*$ represents a submatrix that can have elements of any sign. The strict negativity comes from the fact that $u_1 < u_W$.

Partitioning Z in a compatible manner, we have that

$$\mathrm{diag}\left[\left(\begin{array}{c|c|c} Z_{11} & Z_{12} & Z_{13} \\ \hline Z_{21} & Z_{22} & Z_{23} \\ \hline Z_{31} & Z_{32} & Z_{33} \end{array}\right)\left(\begin{array}{c|c|c} - & - & 0 \\ \hline - & * & + \\ \hline 0 & + & + \end{array}\right)\right] = 0.$$

Considering the $(1,1)$ diagonal block of the above product, noting that $Z \geq 0$, we obtain $Z_{11} = Z_{12} = 0$; and considering the $(3,3)$ diagonal block of the above product we obtain $Z_{32} = Z_{33} = 0$. By symmetry, also $Z_{21} = 0$ and $Z_{33} = 0$.

From here we can easily derive a contradiction. Indeed, if the middle block is nonempty, the matrix is reducible; and if the second block is empty, it is periodic.

Appendix D. Proof of Theorem 12

Let us adopt the convention that when $a = (a_1, \ldots, a_n)$ is a vector, we will understand e^a to apply to it elementwise, i.e., $e^a = (e^{a_1}, \ldots, e^{a_n})$. We will find it convenient to do our analysis in terms of the variables $z(t)$ defined through the relation $x(t) = e^{z(t)}$. In terms of these variables, we can rewrite Eq. (4) as

$$e^{z(t+1)} = P_{\mathcal{C}}\left[e^{z(t) - \alpha \nabla_t}\right]. \tag{15}$$

Observe that projection of a vector a onto the cube $\mathcal{C}$ simply thresholds each a_i between κ and K. Inspecting Eq. (15), we therefore see that we can move the projection inside the exponentiation if we instead project onto the cube $\Omega = [\ln \kappa, \ln K]^W$:

$$e^{z(t+1)} = e^{P_\Omega[z(t) - \alpha \nabla_t]},$$

or

$$z(t+1) = P_\Omega \left[z(t) - \alpha \nabla_t \right]. \tag{16}$$

The trick that makes the proof possible is that we can construct a function so that Eq. (16) becomes a projected gradient descent iteration. To that end, we define

$$g_\Delta(z) = \frac{1}{2} \sum_{i,j=1}^{W} N_{ij} e^{z_i + z_j} - \sum_{i=1}^{W} z_i \sum_{j=1}^{n} N_{ij}(s_i s_j + \Delta_{ij}).$$

The key observation is that the gradient of this function is

$$\begin{aligned} [\nabla g_\Delta(z(t))]_i &= \sum_{j=1}^{W} N_{ij} e^{z_i(t)+z_j(t)} - \sum_{j=1}^{n} N_{ij}\left(s_i s_j + \Delta_{ij}\right) \\ &= \sum_{j=1}^{W} N_{ij}(e^{z_i(t)+z_j(t)} - s_i s_j - \Delta_{ij}) \\ &= [\nabla_t]_i, \end{aligned}$$

where the last step used the definition of $z(t)$, i.e., $e^{z_k(t)} = x_k(t)$ for all $k = 1, \ldots, W$.

Thus we have that

$$z(t+1) = P_\Omega \left[z(t) - \alpha \nabla g_\Delta(z(t)) \right]. \tag{17}$$

Because we now have a projected gradient descent iteration on a convex function (it is, of course, immediate that $g_\Delta(z)$ is convex), it should now be clear that an analysis of this iteration in terms of $z(t)$ id possible, provided by we can upper bound the condition number of the function $g_\Delta(z)$ over the region $z \in [\ln \kappa, \ln K]^W$. To analyze this condition number, we argue as follows.

First, because $\nabla^2 g_\Delta(z) = L_s((N_{ij}/2)e^{z_i+z_j})$, where $L_s(w_{ij})$ refers to the signless Laplacian with weights w_{ij}, i.e.,

$$L_s(w_{ij}) = \sum_{i,j=1}^{W} w_{ij}(e_i + e_j)(e_i + e_j)^T.$$

It follows that $\lambda_{\min}(L_s(w_{ij}))$ is a monotonic function of the weights $\{w_{ij}\}$; this then implies that $g_\Delta(z)$ is a μ-strongly convex over Ω, where $\mu = \kappa^2 N_{\min} \lambda_{\min}(L_s)$, where L_s is the signless Laplacian of the unweighted graph corresponding to the interaction matrix N_{ij} and $N_{\min}$ is the smallest positive N_{ij}. We remind the reader that it was shown in Desai and Rao (1994) that $\lambda_{\min}(L_s) \geq 1/W^3$.

Second, we need to argue that $g_\Delta(z)$ has gradient that is L-Lipschitz, along with an estimate for L. To this end, we reprise the argument we used in the proof of Theorem 11 and argue that for any $a, b \in [\ln \kappa, \ln K]^W$ we have:

$$\begin{aligned} ||\nabla g_\Delta(a) - \nabla g_\Delta(b)||_2 &= ||\mathrm{diag}(e^a) N e^a - \mathrm{diag}(e^b) N e^b||_2 \\ &\leq ||\mathrm{diag}(e^a) N (e^a - e^b)||_2 + ||(\mathrm{diag}(e^a) - \mathrm{diag}(e^b)) N e^b||_2 \\ &\leq K ||N||_2 K ||a-b||_2 + K \max_i |a_i - b_i| ||N||_2 K \sqrt{W} \\ &\leq 2||N||_2 K^2 \sqrt{W} ||a-b||_2, \end{aligned}$$

where we used that for vectors $a, b \in [\ln \kappa, \ln K]^W$ we have that

$$\begin{aligned} ||\mathrm{diag}(e^a)||_2 &\leq K \\ ||e^b||_2 &\leq K\sqrt{W} \\ ||e^a - e^b||_2 &\leq K||a-b||_2 \\ ||\mathrm{diag}(e^a) - \mathrm{diag}(e^b)||_2 &\leq K \max_{i=1,\ldots,W} |a_i - b_i| \end{aligned}$$

In conclusion, we may take $L = 2\sqrt{W}||N||_2 K^2$.

Having obtained bounds on L, μ, we next analyze the performance of Eq. (17). Defining,

$$z_\Delta^* := \arg\min_{z \in \Omega} g_\Delta(z),$$

we have that $z(t)$ converges to z^* with the choice $\alpha = 1/L$; as an immediate consequence, we have that $x(t) \to x_\Delta^*$ where

$$x_\Delta^* = e^{z_\Delta^*}.$$

This proves the first assertion of the theorem, that $x(t)$ converges.

To prove the second and third part, we first need to establish the following technical lemma.

Lemma 16 *If s lies in the interior of $[\kappa, K]^W$ and $\max_{i,j} |\Delta_{ij}|$ is small enough, then z_Δ^* lies in the interior of $\Omega = [\ln \kappa, \ln K]^W$.*

Proof We first argue that the unique minimizer of $g_0(z)$ (i.e., when $\Delta = 0$) is $z^* = \ln s$. Indeed, observe that $\nabla g_0(\ln s) = 0$. Moreover, we have already discussed that, over the region $[\ln a, \ln b]^W$, the Hessian of g_0 has eigenvalues lower bounded by $\mu = a^2 N_{\min} \lambda_{\min}(L_s)$. Since the assumption that the graph corresponding to N is connected and non-bipartite renders implies the signless Laplacian L_s is positive definite (again, see Desai and Rao (1994)), we obtain that $g_0(z)$ is strictly convex. Thus $z^* = \ln s$ is the unique minimizer of g_0 over Ω.

It follows that the same property holds for small enough Δ "by continuity." More formally, the argument is as follows. As a consequence of the function that $\ln s$ is the unique minimizer of $g_0(z)$ over Ω, we have that there is a ball $\mathcal{B}$ of positive radius around $\ln s$ such that the function $g_0(z)$ is strictly smaller on $\mathcal{B}$ than it is on any point of the boundary of the cube Ω. It follows that, for small enough Δ, the function g_Δ will also be strictly smaller on $\mathcal{B}$ than on any point on the boundary of Ω. This implies that the minimium of g_Δ occurs in the interior of Ω. ■

Having established this lemma, we now turn to an analysis of the convergence times. We have that that standard results for projected gradient descent on strongly convex functions, with the choice of of step-size $\alpha = 1/L$, we have that (see Theorem 2.4 of Hazan (2016))

$$g_\Delta(z(t)) - g_\Delta(z^*) \leq e^{-t\mu/(4L)} \left(g_\Delta(z(0)) - g_\Delta(z^*) \right), \tag{18}$$

where μ is the strong convexity coefficient. Our next step is to translate this into a convergence rate for $x(t)$.

First, using the mean value theorem, we have that for any two scalars u, v

$$\min\{e^u, e^v\}|u-v| \leq |e^u - e^v| \leq \max\{e^u, e^v\}|u-v|, \tag{19}$$

and we use this in the next sequence of equations:

$$\begin{aligned} ||x(t) - x^*_\Delta||_2^2 &\leq K^2 ||z(t) - z^*_\Delta||_2^2 && \text{By Eq.(19).} \\ &\leq K^2 \tfrac{2}{\mu} \left(g_\Delta(z(t)) - g_\Delta(z^*)\right) && \text{see Hazan (2016) bottom of page 26.} \\ &\leq K^2 \tfrac{2}{\mu} e^{-t\mu/(4L)} \left(g_\Delta(z(0)) - g_\Delta(z^*)\right) && \text{By Eq.(18)} \end{aligned}$$

Now any function $h(y)$ convex over a convex region $\mathcal{R}$ with L-Lipschitz gradient over the same region satisfies (see proof of Lemma 1.2.3 in Nesterov (2004))

$$h(y_1) \leq h(y_2) + \nabla h(y_2)^T (y_1 - y_2) + \frac{L}{2} ||y_1 - y_2||_2^2,$$

for any $y_1, y_2 \in \mathcal{R}$. If y_2 is further chosen to be a point satisfying $\nabla h(y_2) = 0$, then we have

$$h(y_1) - h(y_2) \leq \frac{L}{2} ||y_1 - y_2||_2^2$$

We next apply this to the function g_Δ, which is convex with L-Lipschitz gradient over the region Ω. By Lemma 16, the minimizer z^*_Δ lies in the interior of Ω, and consequently we have $\nabla g_\Delta(z^*) = 0$. Therefore,

$$\begin{aligned} ||x(t) - x^*_\Delta||_2^2 &\leq K^2 \tfrac{L}{\mu} e^{-t\mu/(4L)} ||z(0) - z^*_\Delta||_2^2 \\ &\leq \tfrac{K^2}{\kappa^2} \tfrac{L}{\mu} e^{-t\mu/(4L)} ||x(0) - x^*_\Delta||_2^2 \quad \text{By Eq.(19)} \end{aligned}$$

Since $||x(0) - x^*||_2^2 \leq WK^2$ because $|x_i(0)| \leq K$, we have that it takes

$$\frac{4L}{\mu} \ln \frac{WK^4 L}{\epsilon \mu \kappa^2}$$

iterations until $||x(t) - x^*_\Delta||_2^2 \leq \epsilon$. Since the quantity L/μ scales polynomially in the number of workers W, this proves the second assertion of the theorem, namely that convergence to any ϵ-neighborhood of the limit x^*_Δ occurs in polynomial time. This proves the second assertion of the theorem.

We now turn to the last assertion of the theorem, i.e., the bound on $||x^*_\Delta - s||_2$. For this part, start with the equation

$$\nabla g_\Delta(z^*_\Delta) = 0,$$

which we argued above will hold for small enough Δ, and observe that its consequence is that

$$||\nabla g_0(z^*_\Delta)||_2 = \left|\left| [\sum_{j=1}^{W} N_{ij} \Delta_{ij}]_i \right|\right|_2 \leq \sqrt{W} ||N||_\infty \max_{i,j} |\Delta_{ij}|. \tag{20}$$

Our final step is to argue that this implies z_Δ^* is close to $\ln s$. Indeed, let us define $\phi(t) = \nabla g_0(\ln s + t(z_\Delta^* - \ln s))$. We thus have that

$$\begin{aligned}\nabla g_0(z_\Delta^*) &= \phi(1) - \phi(0)\\ &= \int_0^1 \phi'(u) du\\ &= \int_0^1 \nabla^2 g_0(u)(z_\Delta^* - \ln s)\ du\end{aligned}$$

so, multiplying both sides by $z_\Delta^* - -\ln s$, we obtain

$$(z_\Delta^* - \ln s)^T \nabla g_0(z_\Delta^*) \geq \mu ||z_\Delta - \ln s||_2^2,$$

where, as before, $\mu = \kappa^2 N_{\min} \lambda_{\min}(L_s)$ is a lower bound on the smallest eigenvalue of $\nabla^2 g_0(u)$ when $u \in [\ln \kappa, \ln K]^W$. Now using Cauchy-Schwarz, this implies

$$||\nabla g_0(z_\Delta^*)||_2 \geq \mu ||z_\Delta^* - \ln s||.$$

Putting this together with Eq. (20), we obtain

$$||z_\Delta^* - \ln s|| \leq \frac{\sqrt{W}||N||_\infty}{\mu} \max_{i,j} |\Delta_{ij}|.$$

Finally,

$$\begin{aligned}||x_\Delta^* - s||_2 &= ||e^{z_\Delta^*} - e^{\ln s}||_2\\ &\leq K||z_\Delta^* - \ln s||_2\\ &\leq K\frac{\sqrt{W}||N||_\infty}{\mu} \max_{i,j} |\Delta_{ij}|.\end{aligned}$$

This concludes the proof of the theorem.