# Effective Crowd Expertise Modeling via Cross Domain Sparsity and Uncertainty Reduction

Sihong Xie[†] Qingbo Hu[†] Weixiang Shao[†] Jingyuan Zhang[†] Jing Gao[§] Wei Fan[‡] Philip S. Yu[†]
[†]Department of Computer Science, University of Illinois at Chicago, Chicago, IL, USA
[§]Department of Computer Science, University at Buffalo, Buffalo, NY, USA
[‡]Baidu BigData Lab, Sunnyvale, CA, USA

## Abstract

Characterizations of crowd expertise is vital to online applications where the crowd plays a central role, such as StackExchange for question-answering and LinkedIn as a workforce market. With accurately estimated worker expertise, new jobs can be assigned to the right workers more effectively and efficiently. Most existing methods solely rely on the sparse worker-job interactions, leading to poorly estimated expertise that does not generalize well to a large amount of unseen jobs. Though transfer learning can utilize external domains to mitigate the sparsity, the auxiliary domains can themselves suffer from incomplete information, leading to inferior performance. There is a lack of principled framework to handle the sparse and incomplete data to achieve better expertise modeling. Based on multi-task learning, we propose a framework that uses the knowledge learned from one domain to gradually resolve the data sparsity or incompleteness problem in the other alternatively. Experimental results on several question-answering datasets demonstrate the effectiveness and convergence of the iterative framework.

## 1 Introduction

Workforce and crowdsourcing markets have been serving as the hubs of human resources and transforming the way that workforce are evaluated, sourced and consumed. For example, LinkedIn connects millions of its members to jobs such that qualified workers and jobs become more accessible; crowd of workers on StackExchange can answer millions of questions (seen as jobs here) ranging from programming languages to cooking.

One common and imperative piece of information of these markets is the worker expertise, which helps matching competent workers to suitable jobs. Given the importance of expertise modeling, there has been a large body of research on the subject. For example, in [3], the authors employed the Naive Bayes classifier to predict whether a LinkedIn member has a specific expertise, using millions of features extracted from member profiles. In [23, 27, 42, 39], worker expertise is modeled as latent factors by factorizing the worker-job interaction matrix. In the question-answer applications, some works have proposed to incorporate the scores that workers earned from their answers, using competition-based ranking models [28, 2, 20], or graph-based models such as PageRank and HITS [37, 36, 41, 38].

These existing models, however, are less effective when facing data sparsity. First, graph-based expert ranking models rely on the "competition graphs" that encode the who-wins-who relationships, and the graphs are assumed to be strongly connected (the "Bradley-Terry-Luce" (BTL) assumption) to deliver consistent results [26]. This assumption nonetheless hardly holds in practice due to the sparsity of worker-job interactions, as verified in our experiments. Regression-based methods [28] make no such assumption and estimate expertise via modeling the worker-job responses. However, in practice, most workers have responded to only a small number of jobs, making robust estimation from the sparse worker-job interactions challenging. Various transfer learning algorithms [25, 19, 40] modeled the worker-job matching problem and addressed data sparsity by borrowing knowledge from external domains. These works assumed that the external domain contains sufficient information to be helpful to the target domain. In our problem settings, quite the contrary, the auxiliary domain can themselves suffer from incomplete information that needs to be estimated. Joint modeling of both the sparse and incomplete data in the two domains can be a more promising direction in expertise estimation.

We first propose to address the sparsity in worker-job responses by exploiting inter-worker similarity to encourage expertise sharing among similar workers: the estimation of the expertise of a worker shall take into account of the expertise of his/her neighbors defined by the inter-worker similarity. Multi-task learning [16, 6, 7, 8, 1, 10] share the same high-level idea.

Here the estimation of the expertise of a worker is considered as a single task. Previous works assumed that task similarity is given or can be reliably learned from the target domain, while the problem setting here challenges this assumption: the sparsity in the worker-job responses hinders the reliable estimation of task similarity. Moreover, the incompleteness of the auxiliary data can impede the transferring of task similarity from the auxiliary domain to help multi-task learning in the target domain. As a novel solution, we exploit the special structures of the worker expertise modeling problem and use the estimated expertise to impute the missing values in the auxiliary domain. The imputed auxiliary data can lead to a more accurate estimation of inter-worker relationships as task similarity, which can in turn enhance worker expertise estimation in the target domain. These two steps continue alternatively to gradually improve the performance of both the expertise and missing value estimation problems. The effectiveness and convergence of this iterative procedure are confirmed on three real-world question-answering datasets.

## 2 Notations and Preliminaries

We summarize the major notations in Table 1. For the matrix $A$, $A_{i:}$ denotes the $i$-th row of $A$ and $A_{:j}$ the $j$-th column of $A$. Suppose that there are $K$ different skills and $M$ workers, then the to-be-estimated worker expertise can be modeled by the $K \times M$ matrix $B$, where each column is a $K$-dimension expertise vector for a worker. A worker can interact with a certain number of jobs, and the scores that the $j$-th worker gained from the interacting jobs are collectively denoted by an $N$-dimension column vector $Y_{:j}$, whose $i$-th entry $(Y_{ij})$ is the score that the worker obtained from the $i$-th job. $Y_{:j}, j = 1, \ldots, M$ comprise the $N \times M$ matrix $Y$. Note that the worker-job interactions are assumed to be sparse, and a large number of entries in $Y$ are missing. The expertise required by the jobs is encoded by the $N \times K$ matrix $X$, where the row vector $X_{i:} \in \mathbb{R}^K$ is the expertise required by the $i$-th job. The $M \times V$ matrix $D$ stores the $V$-dimension auxiliary worker features, with each row being the features for the corresponding worker. We assume that $D$ contains missing values that need to be estimated.

The above formulation is quite general, covering several important real-world applications. For example, at LinkedIn, the required skills of the $i$-th position is given by $X_{i:}$. The $j$-th member who ever took that position can be seen as a worker interacting with the job, and the duration for which the member occupied that position, or any measurable achievements obtained in the position can serve the score $Y_{ij}$. In this paper, $X$ is assumed to be fixed for two reasons. First, in real-world running systems like LinkedIn, these data are built offline and should be stable, since building an interpretable and useful expertise representation requires expensive online experiments [3]. Second, simultaneously inferring both expertise representations and worker expertise, as has been done in matrix factorization, can introduce too many parameters and aggravate the sparsity issue, as we show in the experiments.

Table 1: Notations

| Symbol | Meaning |
|---|---|
| $\mathcal{U}$ | Set of workers |
| $M$ | number of workers |
| $\mathcal{T}$ | Set of tasks |
| $N$ | number of tasks |
| $K$ | number of expertise |
| $\mathcal{R}$ | Set of worker-job responses |
| $D \in \mathbb{R}^{M \times V}$ | User feature matrix |
| $G \in \mathbf{R}^{M \times M}$ | User correlation matrix |
| $Y \in \mathbb{R}^{N \times M}$ | User-task response matrix |
| $X \in \mathbb{R}^{N \times K}$ | Task-expertise matrix |
| $B \in \mathbb{R}^{K \times M}$ | Users expertise matrix |
| $\lVert \cdot \rVert_F$ | Frobenius norm of a matrix |
| $[m]$ | the set $\{1, \ldots, m\}$ |

**2.1 Preliminaries** In this section, we first categorize previous expertise estimation methods into several families, including a regression-based approach, based on which we propose our framework in the next section.

**2.1.1 Latent factor based methods** Methods in this category infer latent representations of workers and jobs as expertise from job features. For example, the authors in [23, 27] adopted language models including bag-of-words, latent Dirichlet allocation (LDA), and non-negative matrix factorization (NMF) [35] for this purpose. Topics of jobs can be seen as the required expertise, and the expertise of a worker is the mixture of the expertise of the jobs that the worker has interacted with. The drawback of such methods is that the relative expertise proficiency, expressed by the scores in the worker-job interaction data, is ignored, and thus expertise cannot be accurately estimated.

**2.1.2 Competition graph based methods** Approaches in this category first build a graph of workers to encode the who-wins-who relationships. For example, in [2, 20, 28], an edge from worker $i$ to $j$ ($j$ won $i$) if $j$ answered a question asked by $i$, or $j$ provided a better answer than that provided by $i$ under the same question. Then expertise level can be estimated based on these competition relationships, using learning to rank [20], Bayesian model TrueSkill [11, 22] and centrality measures [2]. For example, PageRank computes the ranks of nodes on the graph as expertise level, and RankSVM can find expertise to fit the observed competition relationships. Due to data sparsity, however, a fairly large number of workers have only responded to a few jobs, and they will not have sufficient competitions to infer their expertise stably. Further, the competition graph might not have strongly connectedness, which is required to ensure the consistency of the inferred expertise using the PageRank or other ranking algorithms [26]. We empirically demonstrate these observations in the experiments.

#### 2.1.3 Hybrid methods

Methods in this category consider both job features and worker-job interaction scores, where the job features can mitigate the sparsity in the worker-job interactions. For example, in [36], the expertise is first inferred as topics from texts and worker-job interaction scores, then a competition network is built to infer worker expertise level. They did not address the sparsity issue in the two sources, and the constructed answerer-asker competition network again suffers from the sparsity of the worker responses.

#### 2.1.4 Expertise estimation as linear regression

We will based our framework on linear regression to model worker-job interaction scores while considering worker relationships inferred from an auxiliary domain. Given the worker-job interactions and the expertise associated with the jobs, linear regression has been proposed to find the worker expertise [28]. Considering the estimation of the expertise of a single worker. Let the responses of the worker to $N$ tasks given by a column vector $\mathbf{y} \in \mathbb{R}^N$, and the expertise required by the jobs by $X \in \mathbb{R}^{N\times K}$, then the worker expertise, denoted by $\boldsymbol{\beta}$, can be estimated via $\ell_2$-regularized linear regression:

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^K} \frac{1}{2}\|\mathbf{y} - X\boldsymbol{\beta}\|_2^2 + \frac{1}{2}\lambda\|\boldsymbol{\beta}\|_2^2. \tag{2.1}$$

The estimation of $\boldsymbol{\beta}$ for *all* workers can be formulated in a compact way [28]:

$$\min_{B\in\mathbb{R}^{K\times M}} \frac{1}{2}\|Y - XB\|_2^2 + \frac{1}{2}\lambda\|B\|_F^2 \tag{2.2}$$

where $B = [\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_M] \in \mathbb{R}^{K\times M}$. $\|B\|_F^2$ is the square of the Frobenius norm of $B$. A closed-form solution is:

$$B^* = (X^\top X + \lambda I)^{-1} X^\top Y. \tag{2.3}$$

Regression-based approaches do not assume a strongly connected competition network that is required by the competition network based and hybrid methods. However, the matrix $Y$ is usually sparse, since most workers only interact with a small number of jobs (see Figure 1(b) in the experiments). A worker who has insufficient interactions would have poorly estimated expertise that does not generalize to unseen data in prediction very well.

## 3 Proposed Framework

Based on the above regression model, we propose an approach to handle the sparse worker-job interactions with the help of an auxiliary domain, which is assumed to be complete. We encode worker similarity using the auxiliary data (Section 3.1), to guide worker expertise estimation in the target domain (Section 3.2). We investigate missing values in the auxiliary data later.

### 3.1 Construction of inter-worker graphs from auxiliary data

Besides the scores in the worker-job interaction data $Y$ and job expertise specification $X$, workers can leave an extra trail of footprints in the applications. For example, on LinkedIn, the footprints include social connections, current and past job titles, companies/organizations s/he has worked for, etc. Such data are considered as auxiliary data in addition to $X$ and $Y$. These auxiliary data can be of high-dimensional (millions in [3]) if we wish to use a comprehensive set of worker profiles. We adopt dimension reduction to embed workers in a lower dimensional space where worker similarity can be estimated more robustly. Let $D \in \mathbb{R}^{M\times V}$ be the matrix consisting of the auxiliary data, with each row being the features characterizing a worker. We decompose $D$ using SVD: $D = U\Sigma W$, where $U \in \mathbb{R}^{M\times V_0}, V_0 \ll V$. The inter-worker relationships can be represented by the $M \times M$ matrix $G = UU^\top$. $G_{ij}$ measures the correlation between the $i$-th and $j$-th workers, where a significant positive/negative value indicates a strong correlation/anti-correlation, and a small absolute value indicates a weak relationship. $G$ can be represented by a graph with $M$ nodes of workers, and $G_{ij}$ being the weight of the edge $(i, j)$.

### 3.2 Incorporating inter-worker relationships via graph-based multi-task regression

The worker relationships $G$ can help resolve the sparsity in the target domain when estimating the worker expertise using linear regression Eq. (2.2), with a graph-based regularization term enforcing the fusions of worker expertise:

$$\min_{B\in\mathbb{R}^{K\times M}} \frac{1}{2}\|Y - XB\|_F^2 + \lambda\Omega_g(B). \tag{3.4}$$

where $\Omega_g(B)$ is the graph-guided fusion term:

$$\Omega_g(B) = \sum_{(i,j)\in G} |G_{ij}|\|\boldsymbol{\beta}_i - \mathrm{sgn}(G_{ij})\boldsymbol{\beta}_j\|_1. \tag{3.5}$$

Here $\mathrm{sgn}(x)$ returns the sign of the scalar $x \in \mathbb{R}$, and $G$ is the graph encoding the inter-worker relationships defined in Section 3.1. $H$ is an $M \times |E|$ matrix:

$$H_{k,e} = \begin{cases} G_{kj} & \text{if } e = (i,j) \text{ and } k = i \\ -G_{kj} & \text{if } e = (i,j) \text{ and } k = j \\ 0 & \text{otherwise.} \end{cases} \tag{3.6}$$

The effect of $\Omega_g$ is to encourage two workers who are similar based on the auxiliary data to have similar expertise, and dissimilar workers to have disparate expertise. The scalar $|G_{ij}|$ decides how much such effect $G_{ij}$ exerts on the vectors $\boldsymbol{\beta}_i$ and $\boldsymbol{\beta}_j$.

The original worker graph contains $O(M^2)$ edges connecting all worker pairs. Such a dense graph can be noisy and slow down the optimization algorithm solving Eq. (3.4). One can set a threshold to cut off the edges with insignificant absolute edge weights. However, if too many edges are cut off, the inter-worker relationships would vanish too much and Eq. (3.4) becomes Eq. (2.2). According to our experiments, the more edges we keep (in the range of 15% to 25% of the total edges), the better the performance. Therefore we keep 25% of the edges with the largest magnitudes. Note that the matrix $D$ is assumed to capture all possible information such that $G$ is a good estimation of the true worker relationships. We will drop this assumption in Section 4.

**3.3 Optimization algorithm** The optimization objective Eq. (3.4) is convex. However, the difficulties come from the non-smooth and non-separable graph-guided fusion term, which is a common challenge shared by many structured sparse models, such as graph-guided fused lasso and overlapping group lasso [12, 15]. Although interior point method can solve these problems, more efficient algorithms are proposed to achieve faster convergence rate and reduce computational complexity [13, 9, 31, 21, 24]. In the most recent work [10], the authors showed that an ADMM re-formulation of the optimization problem can estimate the worker expertise in a parallel way. Therefore, the proposed expertise model can be scaled to large problems. In the experiments, we adopt the method in [6] as our optimization solver. The dual of the fusion penalty can be written as

$$\Omega_g(B) = \|BH\|_1 = \max_{A \in \mathbb{R}^{K \times |E|} : \|A\|_\infty \leq 1} \langle A, BH \rangle \,. \tag{3.7}$$

This dual norm is still non-smooth, and a smooth lower bound of the dual norm is

$$f_\mu(B) = \max_{A \in \mathbb{R}^{K \times |E|} : \|A\|_\infty \leq 1} \left\{ \langle A, BH \rangle - \frac{\mu}{2} \|A\|_F^2 \right\}, \tag{3.8}$$

where $\mu$ is the smoothing parameter that controls the smoothness of $f_\mu(B)$. Therefore, the following function gives a smooth lower bound of the original objective function Eq. (3.4):

$$\tilde{f}(B) = \frac{1}{2} \|Y - XB\|_F^2 + f_\mu(B). \tag{3.9}$$

The gradient of $\tilde{f}(B)$ is

$$\nabla \tilde{f}(B) = X^\top (XB - Y) + \nabla f_\mu(B), \tag{3.10}$$

where

$$\nabla f_\mu(B) = A^* H^\top, \quad A^* = S(BH/\mu). \tag{3.11}$$

Here $S(x) = x$ if $x \in [-1, 1]$ and $\mathrm{sgn}(x)$ otherwise. $\mu \geq 0$ trades off between the smoothness of $f_\mu(B)$, measured by the Lipschitz constant of $\nabla f_\mu(B)$: $L_\mu = \frac{1}{\mu} \|H\|_2^2$. Let $L_U = \lambda_1(X^\top X) + \frac{2\lambda^2 \max_k d_k}{\mu}$, where $\lambda_1(A)$ is the maximal eigenvalue of $A$ and $d_k = \|G_{\cdot k}\|_2^2$.

**Algorithm 1** Accelerated Gradient Descent

```
**Input**: $X$, $Y$, $G$, MaxIterNum
**Initialization**: $W^0 = \mathbf{0}$
**for** $s = 1 \to$ MaxIterNum **do**
    compute $\nabla \tilde{f}(W^s)$.
    gradient descent: $B^s = W^s - \frac{1}{L_U} \nabla \tilde{f}(W^s)$.
    $Z^s = -\frac{1}{L_U} \sum_{i=0}^{s} \frac{i+1}{2} \nabla \tilde{f}(W^s)$.
    $W^{s+1} = \frac{s+1}{s+3} B^s + \frac{2}{t+3} Z^s$.
**end for**
Output $B^s$.
```

We can borrow the following theorem from [6] for the convergence of the above optimization algorithms.

Theorem 1. *If we require the objective function at $B^s$ to be $\epsilon$ close to the minimum of the objective function, then $O(1/\epsilon)$ iterations are needed.*

The significance of the theorem is that the accelerated gradient descent procedure converges faster than the subgradient method, which has rate $O(1/\epsilon^2)$.

## 4 Joint estimations of inter-worker graph and expertise

Here we assume that the worker profile matrix $D$ is incomplete. This is usually the case in real world applications. For example, a LinkedIn member can only input a small number of important past positions and projects, and not all the details are recorded; on Stackoverflow, due to limit time or interests, an expert could have answered more questions than he/she has answered. The missing information will lead to insufficient workers profiles, and in turn the estimated inter-worker relationships also suffer. If one can complete the missing information to some extent, we shall have a more accurate inter-worker relationship matrix $G$, which can further lead to a more accurate worker expertise estimations via the graph-fused linear regression model. What's even better, if we can utilize the improved expertise estimations to enhance the missing information completion procedure, the prediction performance is likely to move towards a good direction.

We present an iterative algorithm, shown in Algorithm 2, to implement the above idea. Although we assume the special structures of StackExchange dataset, the main idea applies to a wide range of other situations, where worker-job predictions can help worker profile completion. After running Algorithm 1, we have the worker expertise estimation $\hat{B}$ to predict the worker responses to the *training* jobs. Thus the possibly missing responses in the training target $Y$ can be imputed using the corresponding entries in $X\hat{B}$, where $X$ is the expertise required by the training jobs. However, not all predicted values are useful, since a worker will usually respond to only a small number of jobs. We choose to impute the missing entries in $Y$ using the corresponding values in $X\hat{B}$ with the highest predicted responses. However, if too many such missing entries are imputed, a large amount of noisy entries may be introduced. Another question is what value ($\tau$ in Algorithm 2) to use for imputation. We study the sensitivity of the algorithm to these two parameters ($\tau$ and $k$) in the experiments, along with the convergence of the algorithm. Denote the imputed worker-job response matrix by $\hat{Y}$, we can then calculate a new feature vector for a worker by summing up the feature vectors of the jobs (in the matrix $F$) that are assigned to the worker according to $\hat{Y}$.

**Algorithm 2** Iterative Graph-fused Least Mean Square

```
Input: $X$, $Y$, $D$, $F$, $\tau$, $k$, MaxIterNum
Initialization: $D = U\Sigma V^\top$, $G = UU^\top$.
for $s = 1 \to$ MaxIterNum do
    Run Algorithm 1 with current $G$ to obtain $\hat{B}^s$.
    Select the top $k$ entries from each row of $X\hat{B}$ with the largest magnitudes.
    Impute the missing entries with value $\tau$, denote the imputed $Y$ by $\hat{Y}$.
    Re-compute the worker profile $\hat{D} = \hat{Y}F$.
    Re-compute the inter-worker graph $\hat{D} = U\Sigma V^\top$, $G = UU^\top$.
end for
Output $\hat{B}^s$.
```

## 5 Experiments

We demonstrate the effectiveness of the proposed framework using worker-job matching on several question-answer websites. Here a worker is a registered user and a job is a question asked by a user. After worker expertise is estimated using the training data, the goal is to retrieve questions on the test set for the existing users who appear in the training set. Since most users only answered a few questions, the evaluation shall focus on the ranking of the retrieved questions and we use AUC as the evaluation metric.

**5.1 Datasets** Three question-answering (QA) websites from the StackExchange system are adopted: cstheory, unix and english. Given the raw data, stop words were pruned and we extracted the bag-of-word features from the processed texts, with tf-idf transformation applied to the question-word matrices. We use the features of the questions that a worker has responded to as his/her auxiliary profile (as a row in $D$). Note that even we have observed all the worker responses, there are certain missing questions that a worker failed to response. We consider such questions as missing responses for the workers and try to fill up these missing responses while estimating worker expertise.

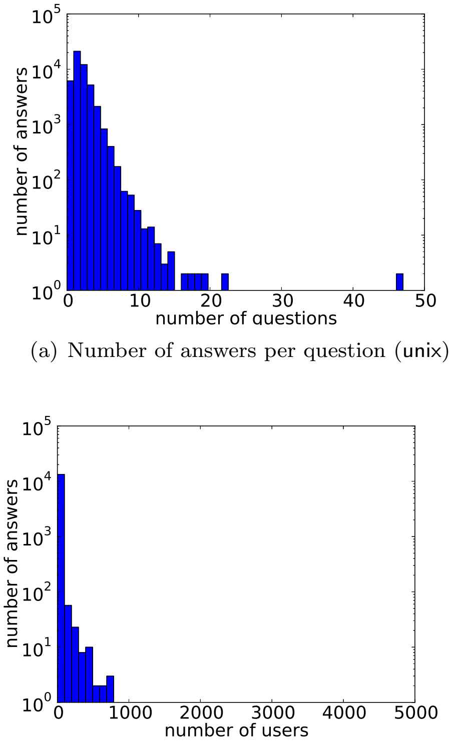
We ran LDA using the GibbsLDA++ package [1] on the question-term matrices to extract latent factors as the skills/expertise required by the jobs (i.e., the matrix $X$). The number of skills is fixed at 200 and other parameters of LDA are set to default values. For practical applications, it is non-trivial to build standardized skill sets [3], which are left to the practitioners' discretion. We selected users who has answered at least 2 questions into the worker-job response matrix, and randomly split the questions into 3 equal-sized disjoint subsets for training, validation and testing. The various dimensionalities of the datasets are shown in Table 2.

We explore various distribution characteristics of the unix datasets in Figure 1 (the other datasets exhibit similar characteristics) and get the following observations. First, the majority of the questions/users have only a small number of answers. For example, only 2311 out of 6149 questions got at least 2 answers on cstheory. If we consider two workers who answered the same question as competitors for that question, the constructed worker competition network would be highly disconnected, as evidenced by the large number of strongly connected components (# of SSCs) in Table 2. Thus the BTL assumption [26] made in [2, 20, 28] is not satisfied. Second, if the simple regression-based approach is adopted (see Eq. (2.2)), for a worker who only answered a few questions, the estimated worker expertise would concentrate on that associated with the answered questions. This is undesirable as the generalization power of the estimated expertise is quite limited.

[1] gibbslda.sourceforge.net

Table 2: Datasets

| | cstheory | unix | english |
|---|---|---|---|
| # Workers | 707 | 4632 | 4909 |
| # Jobs | 4580 | 40011 | 39505 |
| # Responses | 8545 | 56819 | 79961 |
| # Train | 1851 | 16056 | 15912 |
| # Valid | 1392 | 11954 | 11798 |
| # Test | 1337 | 12001 | 11795 |
| # of SCCs | 548 | 2789 | 3766 |



(a) Number of answers per question (unix)

(b) Number of answers per user (unix)

Figure 1: Number of responses per job/worker

**5.2 Baselines** We employed baselines that consider different perspectives of expertise estimation.

- LDA: we use the latent factors obtained from LDA as expertise required by the jobs, and aggregate the expertise of the jobs that a worker has responded to as the worker's expertise. This baseline does not consider the quality of worker responses $Y$.

- LR: for each worker, we estimate his/her expertise based on the observed $X$ and the worker's responses (binarized) $Y$, using logistic regression implemented by the LibLinear package.

- RegLMS: we apply the regularized least mean square regression (Section 2.1.4). This baseline can not only consider the quality of the worker responses to tasks ($Y$), but also the dependency among skills ($(X^\top X + I)^{-1}$). However there is no information sharing among workers, and this baseline can suffer from the sparsity in $Y$.

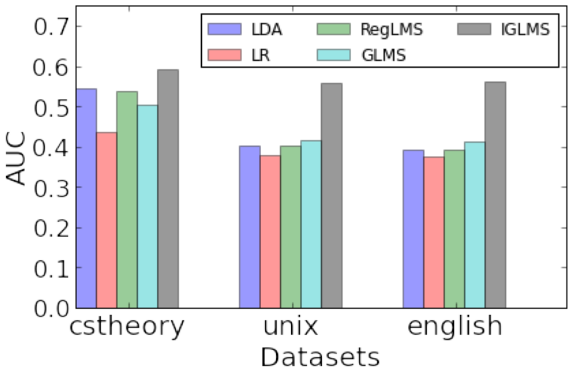
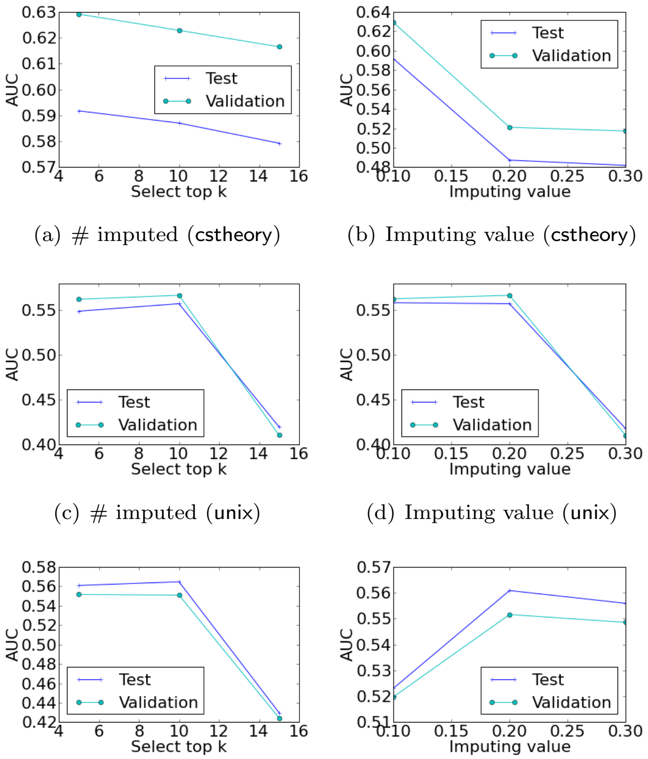


Figure 2: overall performance comparisons

- GLMS: we use the graph-fused lasso (Section 3.2). This baseline incorporates the inter-worker relationships into RegLMS to encourage information sharing and address the sparsity problem. However, it assumes that the inter-worker relationships can be estimated reliably from the worker feature matrix $D$, which may be incomplete.

The proposed algorithm is IGLMS, which brings a further improvement to GLMS by iteratively estimating worker expertise and predicting missing worker-job responses for a more accurate inter-worker graph. Note that we focus on modeling worker expertise to match workers to *unseen* tasks, which cannot be modeled by matrix factorization based approaches.

**5.3 Overall Performance** We compare the performance of the IGLMS algorithm and the baselines in Figure 2. Validation set is used to select the best imputation parameters (see sensitivity studies). IGLMS outperforms all baselines on all datasets, and is 34% better than the runners-up on the last two datasets. The worst baseline across all dataset is LR, which utilizes the least information in the data. LDA and RegLMS have almost the same performance, and we conjecture that both of them use about the same amount of information: the topic distribution of jobs and the worker-job associations in the training data. GLMS is not very stable. It outperforms the other baselines in the last two datasets, indicating the usefulness of the inter-worker relationships $G$. However, it is the second worst algorithm in the first dataset. By jointly modeling the missing responses and worker relationships, IGLMS enhances GLMS and has the best performance.



(a) # imputed (cstheory) (b) Imputing value (cstheory)

(c) # imputed (unix) (d) Imputing value (unix)

(e) # imputed (english) (f) Imputing value (english)

Figure 3: Sensitivity of IGLMS to the number of imputed missing responses and imputing value

**5.4 Parameter Sensitivity** IGLMS has two parameters: how many missing worker-job responses to impute using what values. Figure 3 plot the AUC performance of the algorithm with varying parameters on three datasets. Figures in the first column shows the sensitivity of AUC to the number of imputed missing responses (5, 10 and 15), and those in the second column shows the sensitivity to the imputing value (0.1, 0.2 and 0.3). From the figures, we can see that imputing only 5 missing values achieves good results over all three datasets. The reason is that if too many missing values are filled up, noise can be introduced. On the other hand, the best imputing value can vary across datasets. Fortunately, as the figure shows, the performance varies consistently over the test and validating sets, and we can use the validation set to select the best value for the parameter. In fact, the validation and testing performances are highly correlated across all parameters and datasets, as shown in Figure 4: each point in the plot is a pair of valid-test (corresponding to the $x - y$ axes) performance, and as the validation AUC goes up, the test AUC goes up too.

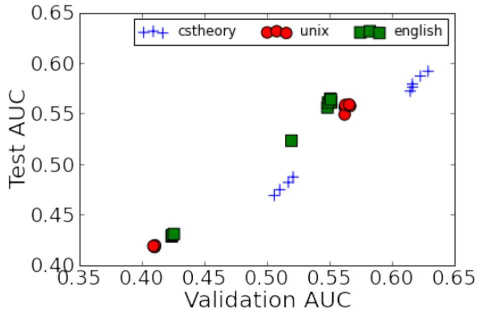


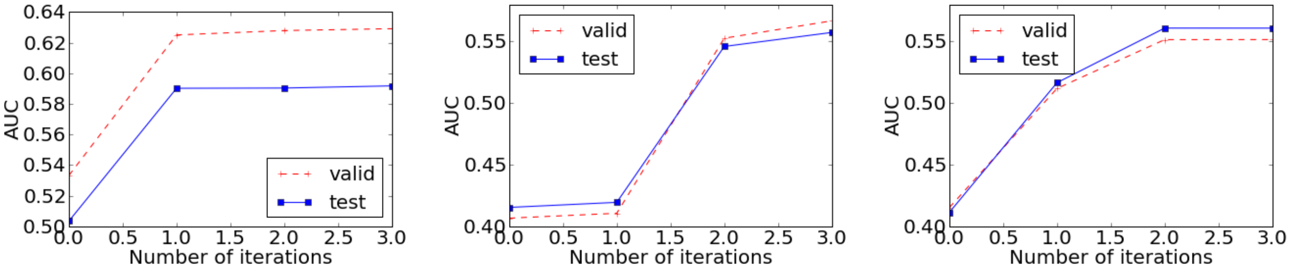
Figure 4: Validation AUC vs. test AUC

**5.5 Convergence** In Figure 5, we plot the performance of IGLMS on the validation and test sets as the algorithm iterates. One can see that the algorithm converges in 4 iterations, and the performances on both the validation and test sets go up as the algorithm iterates on all 3 datasets. These observations confirm the convergence of IGLMS.

## 6 Related Works

Earlier works adopted an information retrieval approach [23, 27], where various language models including TF-IDF, bag-of-words and LDA were used to compute the relevances between jobs and workers. Later on, more information are incorporated in the language modeling. In [42], the authors jointly modeled answer relevance and quality for question routing. Recent works utilized the competition relationships among participants for expertise estimation [41, 38, 36, 2, 20, 28]. The idea is that, the answer scores and the best answer flags can serve as signals that one answerer (worker) is better than the other, and such partial orderings can be translated into relative expertise competences. For example, in [20], the authors proposed to use ranking SVM and TrueSkill [11] to model user expertise using competitions. In [28], an iterative model was proposed to jointly learn the topics of questions and worker expertise, using texts and votings of the QA system. As we analyzed before, these approaches would fail due to the sparsity of the competition network.

There are works on how to exploit the social network properties of crowdsourcing platforms to find experts. For example, in [5], the authors formulated the problem of maximizing the worker confidence (accuracy) and minimizing the cost at the same time. However, they assume that the worker expertise is given, while we consider expertise estimation. In [4], the authors proposed to use social network as a media to reach experts, while the expertise estimation relies on simple text matching. There are also methods focusing on expert-finding on social networks with-



(a) cstheory (b) unix (c) english

Figure 5: Convergence of IGLMS

out worker-job interactions. For example, author ranking on bibliographic networks has been studied intensively [30, 33, 32]. Team-formation finds a group of experts for a specific task on social networks [18, 29]. It focuses on the constraints that the team members need to be well-connected and at the same time have the skills required by the task. Social networks can also serve as auxiliary information to the proposed IGLMS framework. In [34], the authors proposed to estimate reviewer expertise in product review on a heterogeneous network using a propagation-based approach.

The graph-guided fusion multi-task regression belongs to a general family of sparse learning algorithms, where the graph-based constraint can be replaced by more general regularizations to consider other types of task relations, including networks [10], disjoint groups [14], overlapping groups [17] and hierarchies [21]. The problem setting in this paper is of independent interests, where the task relations need to be obtained from auxiliary data source with incomplete information. We exploit both data sources to develop an alternative optimization framework.

## 7 Conclusions and future work

In this paper we study the important problem of expertise estimation in crowdsourcing and workforce markets. We point out the drawbacks of several families of previous methods, such as sparsity in the responses and incomplete information in the auxiliary data source. We propose to address the data sparsity issue by formulating the expertise estimation problem as graph-fused multi-task regression, where an inter-task graph is mined from auxiliary data to encourage information sharing among the estimations of expertise of different workers. We further propose an iterative framework to jointly address the data sparsity and incomplete information in both the target an auxiliary domains, such that the output in one domain can improve outcome in the other. Experiments on 3 real-world datasets demonstrate that the proposed framework is quite promising. In the future, we plan to mine and incorporate more complicated but useful inter-worker relationships from more auxiliary data in the framework.

## Acknowledgment

This work is supported in part by NSF through grants III-1526499, CNS-1115234, and OISE-1129076, and Google Research Award.

## References

[1] Andreas Argyriou, Stéphan Clémençon, and Ruocong Zhang. Learning the Graph of Relations Among Multiple Tasks. Research report, 2013.

[2] Çiğdem Aslay, Neil O'Hare, Luca Maria Aiello, and Alejandro Jaimes. Competition-based networks for expert finding. SIGIR, 2013.

[3] Mathieu Bastian, Matthew Hayes, William Vaughan, Sam Shah, Peter Skomoroch, Hyungjin Kim, Sal Uryasev, and Christopher Lloyd. Linkedin skills: Large-scale topic extraction and inference. RecSys, 2014.

[4] Alessandro Bozzon, Marco Brambilla, Stefano Ceri, Matteo Silvestri, and Giuliano Vesci. Choosing the right crowd: Expert finding in social networks. EDBT, 2013.

[5] Caleb Chen Cao, Yongxin Tong, Lei Chen, and H. V. Jagadish. Wisemarket: A new paradigm for managing wisdom of online social users. KDD, 2013.

[6] Xi Chen, Seyoung Kim, Qihang Lin, Jaime G. Carbonell, and Eric P. Xing. Graph-structured multi-task regression and an efficient optimization method for general fused lasso. *CoRR*, 2010.

[7] Xi Chen, Qihang Lin, Seyoung Kim, Jaime G. Carbonell, and Eric P. Xing. Smoothing proximal gradient method for general structured sparse regression. *Ann. Appl. Stat.*, 2012.

[8] Carlo Ciliberto, Youssef Mroueh, Tomaso Poggio, and Lorenzo Rosasco. Convex learning of multiple tasks and their structure. ICML, 2015.

[9] John Duchi and Yoram Singer. Efficient online and

batch learning using forward backward splitting. *J. Mach. Learn. Res.*, 2009.

[10] David Hallac, Jure Leskovec, and Stephen Boyd. Network lasso: Clustering and optimization in large graphs. KDD, 2015.

[11] Ralf Herbrich, Tom Minka, and Thore Graepel. Trueskill: A bayesian skill rating system. NIPS. 2007.

[12] Rodolphe Jenatton, Jean-Yves Audibert, and Francis Bach. Structured variable selection with sparsity-inducing norms. *J. Mach. Learn. Res.*, 2011.

[13] Rodolphe Jenatton, Julien Mairal, Francis R. Bach, and Guillaume R. Obozinski. Proximal methods for sparse hierarchical dictionary learning. In *ICML*, 2010.

[14] Zhuoliang Kang, Kristen Grauman, and Fei Sha. Learning with whom to share in multi-task feature learning. In *ICML*, 2011.

[15] Seyoung Kim, Kyung-Ah Sohn, and Eric P. Xing. A multivariate regression approach to association analysis of a quantitative trait network. *Bioinformatics*, 2009.

[16] Seyoung Kim and Eric P Xing. Tree-guided group lasso for multi-task regression with structured sparsity. In *ICML*, 2010.

[17] Abhishek Kumar and Hal Daume. Learning task grouping and overlap in multi-task learning. In *ICML*, 2012.

[18] Theodoros Lappas, Kun Liu, and Evimaria Terzi. Finding a team of experts in social networks. KDD, 2009.

[19] Bin Li, Qiang Yang, and Xiangyang Xue. Transfer learning for collaborative filtering via a rating-matrix generative model. ICML, 2009.

[20] Jing Liu, Young-In Song, and Chin-Yew Lin. Competition-based user expertise score estimation. SIGIR, 2011.

[21] Jun Liu and Jieping Ye. Moreau-yosida regularization for grouped tree structure learning. In *NIPS*. 2010.

[22] Mingrong Liu, Yicen Liu, and Qing Yang. Predicting best answerers for new questions in community question answering. WAIM, 2010.

[23] Xiaoyong Liu, W. Bruce Croft, and Matthew Koll. Finding experts in community-based question-answering services. CIKM, 2005.

[24] Julien Mairal, Rodolphe Jenatton, Francis R. Bach, and Guillaume R. Obozinski. Network flow algorithms for structured sparsity. In *NIPS*. 2010.

[25] Weike Pan, Evan Wei Xiang, Nathan Nan Liu, and Qiang Yang. Transfer learning in collaborative filtering for sparsity reduction. In *AAAI*, 2010.

[26] Arun Rajkumar and Shivani Agarwal. A statistical convergence perspective of algorithms for rank aggregation from pairwise data. In *ICML*, 2014.

[27] Fatemeh Riahi, Zainab Zolaktaf, Mahdi Shafiei, and Evangelos Milios. Finding expert users in community question answering. WWW Companion, 2012.

[28] Jose San Pedro and Alexandros Karatzoglou. Question recommendation for collaborative question answering systems with rankSLDA. RecSys, 2014.

[29] Mauro Sozio and Aristides Gionis. The community-search problem and how to plan a successful cocktail party. KDD, 2010.

[30] Y. Sun and J. Han. *Mining Heterogeneous Information Networks: Principles and Methodologies.* Synthesis Lectures on Data Mining and Knowledge Discovery. Morgan & Claypool Publishers, 2012.

[31] Ryan J. Tibshirani and Jonathan Taylor. The solution path of the generalized lasso. *The Annals of Statistics*, 2011.

[32] Guan Wang, Qingbo Hu, and Philip S. Yu. Influence and similarity on heterogeneous networks. In *CIKM*. ACM, 2012.

[33] Ran Wang, Chuan Shi, Philip S. Yu, and Bin Wu. Integrating clustering and ranking on hybrid heterogeneous information network. In *PAKDD*, 2013.

[34] Sihong Xie, Qingbo Hu, Jingyuan Zhang, Jing Gao, Wei Fan, and Philip S. Yu. Robust crowd bias correction via dual knowledge transfer from multiple overlapping sources. In *BigData*, 2015.

[35] Wei Xu, Xin Liu, and Yihong Gong. Document clustering based on non-negative matrix factorization. SIGIR, 2003.

[36] Liu Yang, Minghui Qiu, Swapna Gottipati, Feida Zhu, Jing Jiang, Huiping Sun, and Zhong Chen. CQArank: Jointly model topics and expertise in community question answering. CIKM, 2013.

[37] Jingyuan Zhang, Xiangnan Kong, Roger Jie Luo, Yi Chang, and Philip S. Yu. NCR: A scalable network-based approach to co-ranking in question-and-answer sites. CIKM, 2014.

[38] Jun Zhang, Mark S. Ackerman, and Lada Adamic. Expertise networks in online communities: Structure and algorithms. WWW, 2007.

[39] Zhe Zhao, Zhiyuan Cheng, Lichan Hong, and Ed H. Chi. Improving user topic interest profiles by behavior factorization. WWW, 2015.

[40] Zhou Zhao, James Cheng, Furu Wei, Ming Zhou, Wilfred Ng, and Yingjun Wu. Socialtransfer: Transferring social knowledge for cold-start cowdsourcing. CIKM, 2014.

[41] Guangyou Zhou, Siwei Lai, Kang Liu, and Jun Zhao. Topic-sensitive probabilistic model for expert finding in question answer communities. CIKM, 2012.

[42] Guangyou Zhou, Kang Liu, and Jun Zhao. Joint relevance and answer quality learning for question routing in community qa. CIKM, 2012.