Estimating Sharer Reputation via Social Data Calibration

Jaewon Yang
Stanford University
jayang@stanford.edu

Bee-Chung Chen
LinkedIn
bchen@linkedin.com

Deepak Agarwal
LinkedIn
dagarwal@linkedin.com

ABSTRACT

Online social networks have become important channels for users to share content with their connections and diffuse information. Although much work has been done to identify socially influential users, the problem of finding "reputable" sharers, *who share good content*, has received relatively little attention. Availability of such reputation scores can be useful for various applications like recommending people to follow, procuring high quality content in a scalable way, creating a content reputation economy to incentivize high quality sharing, and many more. To estimate sharer reputation, it is intuitive to leverage data that records how recipients respond (through clicking, liking, etc.) to content items shared by a sharer. However, such data is usually biased — it has a *selection bias* since the shared items can only be seen and responded to by users connected to the sharer in most social networks, and it has a *response bias* since the response is usually influenced by the relationship between the sharer and the recipient (which may not indicate whether the shared content is good). To correct for such biases, we propose to utilize an additional data source that provides *unbiased* goodness estimates for a small set of shared items, and calibrate biased social data through a novel multi-level hierarchical model that describes how the unbiased data and biased data are jointly generated according to sharer reputation scores. The unbiased data also provides the ground truth for quantitative evaluation of different methods. Experiments based on such ground-truth data show that our proposed model significantly outperforms existing methods that estimate social influence using biased social data.

Categories and Subject Descriptors: H.2.8 [**Database Management**]: Database Applications – *Data mining*
General Terms: Algorithms, experimentation.
Keywords: Sharer reputation, Influential users

1. INTRODUCTION

Social networks have made it seamless to share content among connected users, or simply called *connections*. A large fraction of publishers allow easy sharing of content on major social networks like Facebook, LinkedIn, Twitter and others by instrumenting their pages with social network "buttons". Usually these content sharing events, which we shall refer to as *shares*, are broadcasted to user's first degree connections on the social network. The recipients can respond to the broadcasted content through clicks or social gestures like re-shares, likes and comments. Such social gestures may further propagate content to the connections of the respondents. For example, the network update stream (NUS) on the homepage of LinkedIn (shown in Figure 1) provides each user with a stream of content items shared by her connections (among other types of updates), and she can respond to a shared item by clicking the item to see the content, clicking the "share" or "like" button to broadcast the item further to her connections, or commenting about the item. We call such data *social response data*.

Sharer reputation and item attractiveness. We study the problem of estimating sharer reputation for different content topics. Intuitively, the reputation of a sharer for a topic represents her ability of sharing "good" content on that topic. More precisely, we define sharer reputation as *the propensity of a random user interested in the topic (not necessarily connected to the sharer) to respond positively to a typical content item shared by the sharer* (e.g., through a click, share or like). For ease of exposition, we call the propensity of a random user to respond positively to a content item the *attractiveness* of the item. Examples of applications that can leverage such sharer reputation scores are as follows.

- Recommending reputable sharers for users to follow (e.g., [16]): Reputable sharers can serve as information filters for different topics. Following them allows a user to receive the best content in the corresponding topics from a large number of potential candidates.
- Selecting high quality content curators (e.g., [12]): Reputable sharers can help to identify high quality content in a way that is more scalable than hiring editors to do so and more accurate than machine learning approaches. Giving them proper recognition can further incentivize high quality content sharing.
- Providing features for item ranking: Items shared by reputable users can be ranked higher than other items (after adjusting for other features).

Biases in social response data. Estimating *unbiased* user reputation scores from social response data described above is difficult. While such data allows us to measure how positive the recipients respond to the items shared by a sharer, it is inherently biased.

- **Selection bias**: The content shared by a user is usually only seen and responded to by the *immediate connections* of the sharer. However, the definition of reputation we pursue is based on a random user (interested in a topic), to ensure shared content from reputable sharers is attractive for a typical user (interested in the topic). This selectivity in item consumption

KDD'13, August 11–14, 2013, Chicago, Illinois, USA.
Copyright 2013 ACM 978-1-4503-2174-7/13/08 ...$15.00.

imposed by connections in a social network introduces selection bias in social response data.

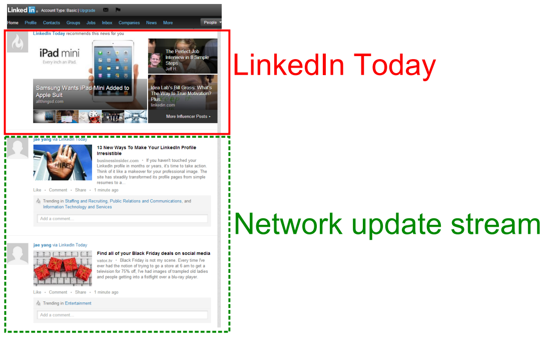
- **Response bias**: The recipients of a shared item *know the sharer's identity* when making a decision to respond. As discussed in studies like [7, 3, 9, 6], users' response in such a social setting is often based on the sharer-recipient relationship, which may not indicate whether shared content is good. For example, friends tend to respond to one another more positively than a random user. Junior people also tend to respond to senior people or their managers more positively [8, 9]. This response bias needs to be corrected.

Prior work. Most work on user reputation or influence in social networks does not try to correct for biases in social data [25, 28, 22, 21]. One common approach is to construct an influence graph among users based on how information propagates (e.g., response to shared items), and then apply methods such as PageRank [4] or HITS [17] to identify influential nodes in the graph. It should be clear that such an influence graph is directly affected by the selection bias and the response bias. Thus, influential notes tend to result from high network in-degree, or high concentration of activities in a sharer's neighborhood, instead of the attractiveness of shared items. Our experiments (Section 4) also show that such an approach fails to estimate sharer reputation. See Section 2 for a discussion of other related work. Bias removal of social data has been studied in question-answering [26, 6] and comment rating [7] environments, where unbiased scores on a small sample of data were provided by human editors. In this paper, we study a different problem setting (i.e., content sharing) and develop a data-driven approach which does not require human supervision.

Our approach. We propose to supplement biased social response data with some *unbiased user action data* that quantifies the attractiveness of a sample of shared items, and estimate unbiased reputation scores by a novel multi-level hierarchical model that describes how the unbiased data and biased data are jointly generated. The unbiased user action data records how users respond (e.g., click) to a sample of shared items that they saw in a way that satisfies the following two important properties:

- **Random user**: Those shared items should be seen by a random sample of users, instead of only the connections of the sharers (to remove the selection bias).
- **Hidden identity**: When a user decides to take an action (e.g., click) on an item, she should not know the identity of the sharer (to remove the response bias).

Availability of unbiased data. We argue that most social network sites can obtain unbiased user action data through active experimentation. Most social network sites either have or can create a content recommendation module on their pages. For example, the LinkedIn Today (LT) module as shown in Figure 1 is such a module that recommends top content items to users, where items can be recommended to all users and sharer identities are not displayed. One way to collect unbiased data is to randomly display a sample of shared items to a random subset of users and record their responses. We note that, even without active randomization, data directly collected from such a module has much less selection and response biases as long as the recommended items cover a set of typical shared items. We also note that the amount of unbiased data does not need to be large, as long as it covers a set of typical shared items. Reasonable performance can be achieved with a few thousands of typical shared items in our experience.



LinkedIn Today recommends this news for you

Samsung Wants iPad Mini Added to Apple Suit
allthingsd.com

The Perfect Job Interview in 8 Simple Steps
Jeff H.

Idea Lab's Bill Gross: What's The Way to True Motivation? Plus...
linkedin.com

jae yang via LinkedIn Today

13 New Ways To Make Your LinkedIn Profile Irresistible

businessinsider.com · If you haven't touched your LinkedIn profile in months or years, it's time to take action. Think of it like a makeover for your professional image. The site has steadily transformed its profile pages from simple resumes to a...

Trending in Staffing and Recruiting, Public Relations and Communications, and Information Technology and Services

jae yang via LinkedIn Today

Find all of your Black Friday deals on social media

vator.tv · Black Friday is not my scene. Every time I've ever had the notion of trying to go a store at 6 am to get a television for 75% off, I've had images of trampled old ladies and people getting into a fistfight over a blu-ray player.

Trending in Entertainment

Figure 1: Snapshot of LinkedIn homepage. LinkedIn Today (LT) module is in the red solid box and Network Update Stream (NUS) module is in the green dashed box.

Modeling challenge. Note that, in a sense, the unbiased user action data is "aggregated" since a user's response to an item in such data does not provide a reputation assessment for a particular sharer, but an aggregated assessment for all of the sharers of that item. On the other hand, the biased social response data is "disaggregated" since each response is an assessment for an individual sharer. The main challenge is how to jointly model *unbiased aggregated* user response on a *small* set of items and a *large* amount of *biased disaggregated* user response on all shared items to obtain unbiased reputation scores.

Contributions. We study a novel problem of estimating *unbiased* sharer reputation scores by combining biased social response data and unbiased user action data through a novel multi-level hierarchical model. Our model combines information by coupling the latent reputation score in the unbiased data with the latent reputation score in the biased data through a linear regression with sharer specific coefficients. To improve model performance with small amounts of unbiased data, we propose a novel co-sharing Markov random field prior for the reputation scores in the unbiased data. We provide rigorous evaluation of our method through ground truth provided by the unbiased data. We show that our method provides significant improvement compared to existing methods that estimate social influence through biased network response data.

2. RELATED WORK

Estimating the importance of individual users in terms of information diffusion [23] has been studied under the topic of finding *influential* users who significantly affect their neighbors (connections). Following the seminal theory on the "influentials" by Katz and Lazarsfeld [14, 29], the problem of identifying influential users has been studied in several different contexts. Before discussing how our work is different from work on user influence, we first note that influential users usually refer to those who affect their neighbors [20], while reputable users in our definition refer to those who receive good response from an anonymous audience.

One of the well studied framework for finding influential users is the influence maximization problem [15] where one chooses a seed set of nodes so that the information from the seed set has maximum reach in the network. While we aim at estimating the reputation (some latent property) for *all* nodes, the influence maximization problem focuses on picking a small number of important nodes. Also, the methodologies for influence maximization have been illustrated mostly with synthetic data, whereas we use unbiased ground-truth data for both training and testing.

Another line of research that has a long history is related to ranking the nodes in a network. Examples of such methods include

PageRank [4] or HITS [17] that identify important nodes from the network structure [13, 18]. Early studies use just network information to estimate users' influence [30], same as PageRank. Recently, as researchers realized that diffusion process may be different from the network structure [5], ground-truth diffusion data as well as network information are used to identify influential users in information diffusion [25]. However, we note that the ground-truth diffusion data may still be heavily influenced by the underlying network structure. For example, if two users are not connected in the network, the current data-driven approaches assume that the two users fail to infect each other in any case. Our method can avoid this problem by calibration against network independent data (responses from anonymous crowd). Furthermore, our evaluation is quantitative as we have ground-truth data for the users' reputation, while the previous methods used manual inspection [22] or qualitative analysis [25].

Another weakly related line of research are empirical studies that analyze the relationship between the reputation and the behavior of users. Question answering sites adopt reputation scoring system where users who give the right answer earn reputation scores as rewards. [2, 27, 6] studied how reputation scores of a user would affect the quality of answers by the user. Our method to estimate latent reputation could also be helpful in designing more effective reputation scoring systems.

3. METHOD DETAILS

In this section, we describe our novel method for estimating sharer reputation through joint modeling of unbiased aggregated response (in the unbiased user action data) and biased disaggregated response (in the biased social response data).

3.1 Problem Setting

We begin with a precise definition of user reputation that we study in this paper.

Sharer reputation. We define the reputation score of sharer s for topic k as *the propensity that a randomly selected user who is "interested in" topic k but not necessarily connected to s would take a positive "action" on a random content item shared by sharer s.* Typically, a user is interested in multiple topics and for simplicity we assume a user's interest affinity to various topics is known. If this is not the case, methods for identifying users' topical interests (e.g., [24, 10, 19]) can be applied before estimating user reputation. Based on the data available to us, we use *click* as our main action. Note that other kinds of actions can be handled in a similar manner.

Our motivation for defining reputation as above is the following: (1) It makes an explicit connection to a common type of objective in content recommendation, i.e., to maximize user actions on the recommended items. For example, recommending highly scored sharers based on this definition of reputation would maximize clicks by respondents. (2) Reputation scores for sharers with "sufficient" amount of "unbiased" data can be computed accurately and provides valuable ground-truth data to evaluate the accuracy of different methods. We will discuss data sufficiency and unbiasedness in more details later.

We now provide a more detailed description of the two kinds of data we use to estimate reputation scores.

Biased social response data. On social networking sites, users share items that in turn propagate through their connections to other users. When a recipient sees a shared item together with the sharer's identity, she may respond to the item through clicking, liking, resharing, etc., or just ignore it depending on factors like item quality, whether the sharer is a close friend, etc. In our experiments, we use log data collected from the Network Update Steam (NUS) module on the homepage of LinkedIn.

Let $z_{sij} \in \{0, 1\}$ indicate user i's response to item j shared by sharer s. For simplicity, we do not distinguish different types of positive response like clicks, likes, re-shares, and just look at whether there is any positive response. This response data can potentially be used to quantify the reputation of the sharer. However, selection and response biases (as defined and discussed in Section 1) need to be corrected.

Unbiased user action data. To help remove biases in social response data, we can collect some unbiased user action data in addition to the biased social response data. Similar to social response data, unbiased user action data records users' response to the items that they saw. However, it needs to satisfy the random user and hidden identity properties defined in Section 1.

Although the unbiased user action data does not have response bias and minimal selection bias, sharing information in some form has to be available to allow estimation of sharer reputation scores. Here, although identities of sharers is not known to the respondents, the system knows the set of sharers for each item, it opens the door to combine this information with biased social response data to estimate sharer reputation scores.

Unbiased user action data can be collected through active experiments on the site or approximated by log data from some item recommendation application. In our experiments, we take the latter approach. Specifically, we use log data from the LinkedIn Today (LT) module on the homepage of LinkedIn. We note that this dataset has no response bias and items can be recommended to all users (not just the sharer's connections), but a different kind of slight selection bias may exists since items are not recommended to users randomly — the serving scheme uses some notion of article popularity. However, because article popularity is typicaly estimated through an explore/exploit algorithm that also ensures some degree of randomization [1], selection bias is weak. See Section 4 for details of our data. As discussed in Section 1, this kind of nearly unbiased user action data is not unique to LinkedIn and can be made available for many social network sites.

Let $y_{ij} \in \{0, 1\}$ indicate user i's response on item j without knowing who shared item j. In our dataset, each click is a positive action. Other types of actions can also be used. Let $\mathcal{J}_s$ denote the set of items shared by sharer s and $\mathcal{S}_j$ denote the set of sharers who shared item j. It is important to note that although such user action data does not have (or is less affected by) the biases found in the social response data, sharing information is available at an "aggregated" resolution which poses additional challenges:

- Such unbiased data is usually *sparse*. For example, in our dataset, the number of items is much smaller than the number of items in the social response data; also, many users share items that never occur in the unbiased user action data.
- The *attribution* of credit for a user action on an item to a sharer of the item is indirect since there are usually multiple sharers for a single item. Because of data sparsity, we are also likely to face a situation where the number of sharers is much more than the number of shared items (this is true for our dataset), this makes the attribution problem even more difficult.

Problem definition. Given a set $\{z_{sij}\}$ of social response data and a set $\{y_{ij}\}$ of unbiased user action data together with a user interest vector $\boldsymbol{\eta}_i$ for each user i, a feature vector $\boldsymbol{x}_i$ for each user i and a feature vector $\boldsymbol{w}_{si}$ for each pair of connected users s and i, the goal is to estimate unbiased reputation score μ_{sk} of each sharer s on each topic k, which intuitively represents the the propensity that a random user who is interested in topic k would take a positive action on a random content item shared by user s.

3.2 Model

We now develop our generative model for estimating topic-specific user reputation scores (μ_{sk}, the reputation of sharer s on topic k), which are unknown *latent factors* to be learned from data. The model fitting algorithm will be described in Section 3.3

User action model. For the unbiased user action data, we assume the mean of the binary response y_{ij} for user i on item j is a function of user i's interest vector η_{ik} for different topics k, and the attractiveness p_{jk} of item j for users interested in different topics k. More specifically,

$$y_{ij} \sim \text{Bernoulli}(\text{probability} = \sigma(\textstyle\sum_k \eta_{ik} p_{jk} + b)), \qquad (1)$$

where $\sigma(x) = 1/(1+e^{-x})$ is the sigmoid function and b is a bias term to be learned from data. Here, p_{jk} is a latent factor associated with (item j, topic k), also to be learned from data. We note that the interest vector η_{ik} is assumed to be given or has been extracted from the user's profile and content consumption patterns in a separate process before our modeling.

Aggregation of user reputation. We connect attractiveness of items to user reputation through modeling the attractiveness p_{jk} of item j for users interested in different topics k as the average of reputation scores μ_{sk} of the sharers $s \in \mathcal{S}_j$ of item j; i.e.,

$$p_{jk} \sim \mathcal{N}(\text{mean} = \frac{1}{|\mathcal{S}_j|} \sum_{s \in \mathcal{S}_j} \mu_{sk}, \ \text{var} = \frac{1}{\lambda_1 |\mathcal{S}_j|}) \qquad (2)$$

where λ_1 is a tuning parameter which represents the strength of a-priori belief that the attractiveness of an item is connected to the reputation scores of the sharers of the item . We note that from the generative point of view, we aggregate user reputation to generate item attractiveness. However, from the estimation point of view, we need to *disaggregate* attractiveness to obtain user reputation.

Co-sharing random-field prior. Recall that the unbiased user action data is sparse. In particular, it is quite likely that the number of items j in the unbiased data is much smaller than the number of sharers s. This is in fact the case in our LT dataset and poses a challenge when we disaggregate attractiveness of a small number of items to obtain reputation scores for a large number of sharers. One common approach is to *shrink* all reputation scores to 0 (intuitively, when we lack data to estimate a score, make sure it is close to a neutral value 0 and does not overfit the data). This kind of shrinkage does not work well in our scenario (see Section 4.2 for experimental results) since we lack data for most of our sharers and hence a large fraction of them will obtain reputation scores close to 0. Hence, we leverage co-sharing data and propose a novel co-sharing Markov random-field prior to regularize the μ_{sk}'s.

The basic idea of our co-sharing random field prior is as follows. In the absence of observed user actions, we assume the reputation score of sharer s is similar to the reputation scores of other users who share the same items as sharer s; this leads to the following Markov random field prior.

$$(\mu_{sk} \mid \{\mu_{tk} : \text{all sharer } t \neq s\}) \sim \text{Normal distribution with}$$

$$\text{mean} = \frac{\sum_{j \in \mathcal{J}_s} \frac{1}{|\mathcal{S}_j|} \sum_{t \in \mathcal{S}_j : t \neq s} \mu_{tk}}{\sum_{j \in \mathcal{J}_s} (1 - \frac{1}{|\mathcal{S}_j|}) + \lambda_2/\lambda_1} \qquad (3)$$

$$\text{var} = \frac{1}{\lambda_1 \sum_{j \in \mathcal{J}_s} (1 - \frac{1}{|\mathcal{S}_j|}) + \lambda_2}$$

To understand this co-sharing random-field prior, note that the neighbors of each sharer s are all other sharers who have shared at least one common item. We also see that the weight on a neighbor depends not only on the number of co-shared items, but also on the total sharers per common item. If co-sharing happens to an item with many other sharers, the weight is discounted.

We now look at the formulation from a technical perspective; we begin with the prior mean that performs double averaging. For a given item j shared by s, we average the reputation scores μ_{tk} of other users t who also share item j. Then, we average over all the items shared by s. Note that the actual averaging in the equation is a bit special. For a given item j, instead of dividing the sum of μ_{tk} by $|\mathcal{S}_j| - 1$ (which is the number of elements in the sum), we divide the sum by $|\mathcal{S}_j|$ and correct this bias using $\sum_{j \in \mathcal{J}_s} (1 - \frac{1}{|\mathcal{S}_j|})$ in the denominator. We choose this kind of averaging because it provides a very clean prior probability density function (see Equation 4). We shall denote the co-sharing Markov random field prior as $\Pr^{\text{CSH-MRF}}(\{\mu_{sk}\})$.

PROPOSITION 1. *The joint log-prior distribution* $\log(\Pr(\{p_{jk}\}, \{\mu_{sk}\}) = \log(\Pr(\{p_{jk}\} \mid \{\mu_{sk}\}) \cdot \Pr^{\textit{CSH-MRF}}(\{\mu_{sk}\}))$ *is*

$$-\frac{\lambda_1}{2} \sum_s \sum_{j \in \mathcal{J}_s} (p_{jk} - \mu_{sk})^2 - \frac{\lambda_2}{2} \sum_s \mu_{sk}^2 + \textit{constant} \qquad (4)$$

The proof follows by completing squares and performing routine algebraic computations. From the above formula, we can clearly see the roles of tuning parameters λ_1 and λ_2 when we estimate p_{jk} and μ_{sk} by maximizing the log-posterior function. λ_1 specifies the strength of interaction between item attractiveness p_{jk} and user reputation μ_{sk}, while λ_2 specifies how strongly we want to shrink user reputation toward 0.

If we have a large amount of unbiased user action data that covers items shared by all users, the above model alone would be sufficient. However, unbiased user action data is usually sparse. In particular, it usually covers a small number of items and a small fraction of sharers. Thus, we need to leverage biased social response data but correct for both the selection and response biases through proper modeling.

Social response model. In the biased social response data, we assume that each response z_{sij} represents whether user i would respond positively to item j shared by sharer s, and it is modeled as a function of user i's interest vector η_{ik} and the "uncalibrated reputation score" α_{sk} of sharer s on different topics k; i.e.,

$$z_{sij} \sim \text{Bernoulli}(\text{probability} = \sigma(\textstyle\sum_k \eta_{ik} \alpha_{sk} + \boldsymbol{\beta}' \boldsymbol{x}_{si})), \qquad (5)$$

where $\boldsymbol{x}_{si}$ is a feature vector including features that can potentially explain the bias between users s and i, $\boldsymbol{\beta}$ is a vector of regression coefficients to be learned from data, and α_{sk} is a latent factor also to be learned from data. Here, we call α_{sk} uncalibrated reputation because user behavior in social response data can be quite different from that in the unbiased user action data from which we obtain the unbiased reputation μ_{sk}.

Regression-based calibration. We model the relationship between α_{sk} and μ_{sk} through a linear regression, where the regression coefficients depend on user features; i.e.,

$$\mu_{sk} \sim \mathcal{N}(\text{mean} = (\boldsymbol{\phi}_k' \boldsymbol{x}_s) \alpha_{sk} + \boldsymbol{\theta}_k' \boldsymbol{x}_s, \ \text{var} = 1/\lambda_3), \qquad (6)$$

where $\boldsymbol{x}_s$ is a feature vector of sharer s, and $\boldsymbol{\phi}_k$ and $\boldsymbol{\theta}_k$ are vectors of topic-specific regression coefficients to be learned from data. Here, we calibrate α_{sk} through a linear function and predict the

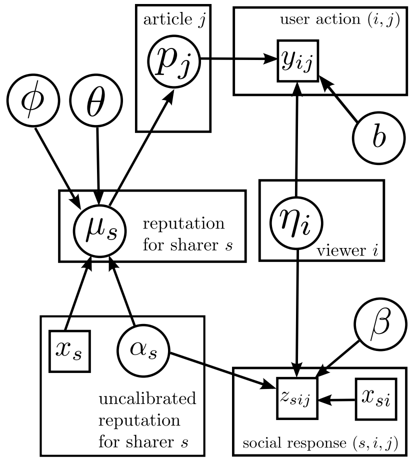


Figure 2: Graphical Representation of our model (variance components are not shown)

Symbol	Description
Observation	
z_{sij}	User i's response to item j shared by sharer s
y_{ij}	User i's response on item j
$\mathcal{J}_s$	The set of items shared by sharer s
$\mathcal{S}_j$	The set of sharers who shared item j
η_{ik}	User i's interest in topic k
$\boldsymbol{x}_s$	Feature vector for sharer s
$\boldsymbol{x}_{si}$	Feature vector between sharer s and user i
Variables to be learned	
μ_{sk}	Unbiased reputation score for sharer s on topic k
α_{sk}	Uncalibrated reputation score for sharer s on topic k
p_{jk}	Item j's attractiveness in topic k
$\boldsymbol{\phi}_k, \boldsymbol{\theta}_k$	Topic-specific regression coefficients between μ_{sk} and α_{sk}
$\boldsymbol{\beta}$	Regression coefficients for a bias term for z_{sij}
b	Bias for y_{ij}

Table 1: Definitions of the symbols.

unbiased reputation μ_{sk} by regressing on α_{sk}. In this regression, $\boldsymbol{\phi}'_k\boldsymbol{x}_s$ is the slope and $\boldsymbol{\theta}'_k\boldsymbol{x}_s$ is the intercept. Different users can have different slopes and intercepts depending on their user features. This provides more flexibility and leads to better performance.

Summary. Figure 2 shows the graphical representation of our model. We also summarize our notations in Table 1. We denote the response in unbiased and biased context as $\boldsymbol{Y}$ and $\boldsymbol{Z}$ respectively. Representing all unknown parameters as Θ, we summarize our model below.

- $\boldsymbol{Y}$ and $\boldsymbol{Z}$ are conditionally independent:

$$\Pr(\boldsymbol{Y}, \boldsymbol{Z}|\Theta) \quad = \quad \Pr(\boldsymbol{Y} \mid \{p_{jk}\}, \{\eta_{ik}\}) \cdot \Pr(\boldsymbol{Z} \mid \{\alpha_{sk}\})$$

- Joint prior on latent variables:

$$\Pr(\{p_{jk}\}, \{\mu_{sk}\} \mid \{\alpha_{sk}\}) = \Pr(\{p_{jk}\}|\{\mu_{sk}\}) \cdot \\ \Pr^{\text{CSH-MRF}}(\{\mu_{sk}\}) \cdot \Pr(\{\mu_{sk}\}|\{\alpha_{sk}\}, \boldsymbol{\phi}_k, \boldsymbol{\theta}_k)$$

where $\Pr^{\text{CSH-MRF}}(\{\mu_{sk}\})$ is the co-sharing Markov random field prior. Note that the prior on $[\{\mu_{sk}\}]$ is proportional to

$$\Pr^{\text{CSH-MRF}}(\{\mu_{sk}\}) \cdot \Pr(\{\mu_{sk}\}|\{\alpha_{sk}\}, \boldsymbol{\phi}_k, \boldsymbol{\theta}_k).$$

3.3 Model Fitting

Although we specify our model using a probabilistic framework, parameter estimation is performed using an optimization approach for the sake of scalability. Our goal is to find the mode of posterior distribution $Pr(\Theta|\boldsymbol{Y}, \boldsymbol{Z})$. We obtain this by maximizing $\log \Pr(\Theta|\boldsymbol{Y}, \boldsymbol{Z})$, which is a non-convex problem but one can obtain the mode by using a coordinate ascent approach. We use coordinate ascent since the set of conditional maximizations we iterate are standard regression problems and could be solved through readily available software.

We have to find the maximum of

$$\begin{aligned} & \text{argmax}_\Theta \log \Pr(\Theta|Y, Z) \\ = \quad & \text{argmax}_\Theta \log \Pr(Y, Z|\Theta) + \log \Pr(\Theta) \\ = \quad & \text{argmax}_\Theta \log \Pr(Y|\Theta) + \log \Pr(Z|\Theta) + \log \Pr(\Theta) \end{aligned} \tag{7}$$

where the last equality comes from the conditional independence between Y and Z given Θ. Our model specifies $\Pr(\Theta)$ as follows:

$$\begin{aligned} \log \Pr(\Theta) \quad = \quad & \log \Pr(\{p_{jk}\}|\{\mu_{sk}\}) + \log \Pr^{\text{CSH-MRF}}(\{\mu_{sk}\}) \\ & + \log \Pr(\{\mu_{sk}\}|\alpha, \boldsymbol{x}, \boldsymbol{\theta}, \phi) \end{aligned} \tag{8}$$

We can express $\log \Pr(\Theta)$ as follows:

$$\begin{aligned} \log \Pr(\Theta) \quad = \quad & -\tfrac{\lambda_1}{2} \textstyle\sum_s \sum_{j \in \mathcal{J}_s} (p_{jk} - \mu_{sk})^2 - \tfrac{\lambda_2}{2} \sum_s \mu_{sk}^2 \\ & -\tfrac{\lambda_3}{2} \textstyle\sum_{s,k} (\mu_{sk} - (\boldsymbol{\phi}'_k \boldsymbol{x}_s)\alpha_{sk} - \boldsymbol{\theta}'_k \boldsymbol{x}_s)^2 \\ & +\text{constant} \end{aligned} \tag{9}$$

We develop a coordinate ascent approach to solve Equation 7 efficiently. We note that if we are to solve for one set of variables (*e.g.*, μ_{sk}s or p_{jk}s) with other sets fixed, then each subproblem becomes a regression problem with regularization. First, we consider fitting p and the corresponding bias b by solving the following problem:

$$\underset{\{p_{jk}\}, b}{\text{argmax}} \log \Pr(Y|\{p_{jk}\}, b, \{\eta_{ik}\}) - \frac{\lambda_1}{2} \sum_{j,k} (p_{jk} - \frac{1}{|S_j|} \sum_{s \in S_j} \mu_{sk})^2 \tag{10}$$

As η_{ik}s are fixed, Y is a logistic function of p_{jk}s and b, and thus this problem is logistic regression with gaussian priors which can be solved efficiently [11].

Second, we aim to fit μ_{sk}s with other variables fixed. Instead of solving the above problem for all μ_{sk}s, we can optimize each μ_{sk} one at a time by solving the following subproblem for each μ_{sk}:

$$\begin{aligned} \text{argmax}_{\mu_{sk}} \quad & (-\lambda_1 \textstyle\sum_{j \in \mathcal{J}_s} (p_{jk} - \mu_{sk})^2 \\ & -\lambda_3 (\mu_{sk} - (\boldsymbol{\phi}'_k \boldsymbol{x}_s)\alpha_{sk} - \boldsymbol{\theta}'_k \boldsymbol{x}_s)^2 - \lambda_2 \mu_{sk}^2) \end{aligned} \tag{11}$$

which is a standard linear regression problem.

Third, we update α_{sk}s by solving:

$$\underset{\alpha, \boldsymbol{\beta}}{\text{argmax}} \log \Pr(Z|\alpha, \boldsymbol{\beta}) + \log \Pr(\alpha|\mu) \tag{12}$$

From Equation 6, we can show $\Pr(\alpha_{sk}|\mu_{sk}, \boldsymbol{\theta}_k, \boldsymbol{\phi}_k)$ is a Gaussian distribution with mean $\frac{\mu^Y_{sk} - \boldsymbol{\theta}'_k \boldsymbol{x}_s}{\boldsymbol{\phi}'_k \boldsymbol{x}_s}$ and variance $\frac{1}{(\boldsymbol{\phi}'_k \boldsymbol{x}_s)^2 \lambda_3}$. Therefore, the problem of updating α_{sk}s is a logistic regression with Gaussian priors.

Fourth, we update $\boldsymbol{\phi}_k$ and $\boldsymbol{\theta}_k$ by fitting linear regression for each topic k as given by

$$\underset{\boldsymbol{\phi}_k, \boldsymbol{\theta}_k}{\text{argmax}} -\lambda_3 \sum_{s,k} (\mu_{sk} - (\boldsymbol{\phi}'_k \boldsymbol{x}_s)\alpha_{sk} - \boldsymbol{\theta}'_k \boldsymbol{x}_s))^2 \tag{13}$$

We iterate the four steps described above until the likelihood converges. In our experiments, our method converges within less than 10 iterations. After learning the parameters, we output the estimates for user s's reputation on topic k as the conditional expectation of μ_{sk} given $(\boldsymbol{\phi}'_k \boldsymbol{x}_s)\alpha_{sk} + \boldsymbol{\theta}'_k \boldsymbol{x}_s$. Note that for cold-start sharers (users who never shared on unbiased context), the estimated reputation score is simply given by $(\hat{\boldsymbol{\phi}}'_k \boldsymbol{x}_s)\hat{\alpha}_{sk} + \hat{\boldsymbol{\theta}}'_k \boldsymbol{x}_s$

4. EXPERIMENTS

In this section, we illustrate our method using LinkedIn datasets. We compare it with two versions of PageRank and a few special cases of our model. We note that availability of unbiased data gives us the ground truth to quantify the performance of various methods.

Dataset description. LinkedIn is the largest professional network in the world with more than 200 million members as of December 31, 2012. The unbiased user action data is obtained from the LinkedIn Today module (LT), while the biased (but granular) social response data is obtained from user interactions with the network update stream (NUS). Figure 1 shows a snapshot of the homepage of LinkedIn and the two modules. The data used in our analysis was collected during a four-month period from May 2012 to August 2012. All unique user identifiers were anonymized.

LinkedIn Today (LT) dataset. A large part of our LT data comes from a random sample of users, to whom LT randomly recommends top algorithmically picked articles that match users' profiles. During the data collection period, LinkedIn Today recommends 3 items (which are articles shared by some sharers) to each user visit. When a user responds to a recommended item in LT, she does not know who shared the item and, thus, can respond in a more unbiased way. Also, LT recommends items to all users (instead of only users connected to the items' sharers) with randomization; thus, user selection bias is weak. To mitigate bias due to displaying same items to a user multiple times, our response y_{ij} for user i on article j is only confined to first view events; i.e., the first time an article is seen by the user. We also only consider users who clicked at least once during the four-month period.

Network Update Stream (NUS) dataset. On LinkedIn, an item (article) shared by any sharer s propagates to the connections of s through NUS; the connections of s can see such a sharing event with the sharer's identity in their NUS modules. The connections can respond by clicking the item or ignore the sharing event. Similar to the LT data, we only consider responses to first view events; i.e., z_{sij} represents whether user i clicked item j shared by sharer s when user i saw item j in her NUS module for the first time.

Users' topical interest. In our experiments, the topical interest η_{ik} of a user i is obtained through the industries (*e.g.*, Internet, Finance, Semiconductor, Communication, and so on) that the user follows or belongs to. On LinkedIn, every user belongs to at least one industry, which is indicated in their profile and can also be inferred through the user's current company. A user can also "follow" a number of industries to obtain article recommendations related to those industries even if she does not belong to them. Using industries to define topics is natural for LinkedIn since it focuses on professional news. Thus, we seek to estimate sharer reputation scores for each industry. Analysis with other kinds of topics can be done similarly. In fact, our model can also be extended to work when η_{ik}s are latent variables to be estimated from data by adding another layer to the hierarchical model, which is future work.

Proxies for ground-truth sharer reputation. As a proxy to measure reputation of sharer s on topic k, we use the average *LT CTR* (Click-Through Rate using the unbiased LT dataset) of items shared by sharer s based on response by viewers of LT interested in topic k. More specifically, for each (sharer s, topic k) pair, we count the numbers of views and clicks by viewers interested in topic k on the set of items shared by s in the unbiased LT data. Then, average LT CTR = number of clicks / number of views.

Evaluation Metric. To evaluate the performance of a model, we split our data into a training set and a test set (the splitting method will be described later), and then train the model using the training set and evaluate the method using the test set. To reduce noise, we compute evaluation metrics based on a set of "test sharers" who satisfy the following two conditions: (1) A test sharer must have at least 10 shared items in the unbiased LT data in the test set; otherwise his/her average LT CTR (our proxy for ground-truth reputation score) would be noisy since it is based on small counts. (2) A test sharer must also have at least 100 view events in the NUS data; otherwise, given the low click rate in NUS, she may receive too few clicks to exploit the NUS data for estimating her reputation due to noise induced by small counts.

Since our goal is to rank sharers based on reputation, we use two ranking-based evaluation metrics. We note that because a ranking-based metric does not depend on the scale of the predicted reputation scores, it is particularly useful for comparing methods that generate scores on different scales. For example, scores from our model and PageRank scores can be on very different scales, but what we focus on is the ranking of sharers. Intuitively, a good method assigns higher scores to sharers who share better items.

- **Kendall's τ**: To determine whether the ranking of sharers based on scores produced by a model is concordant with the ranking of sharers based on their average LT CTR (our proxy for ground-truth reputation scores), we use Kendall's τ rank correlation coefficient [25] (the higher the better).
- **CTR of top k sharers**: The top k sharers identified by a better model should share better items (having higher average LT CTR) than those identified by a worse model. We define the CTR of top k sharers of a model as the average LT CTR of the set of items shared by the top k sharers identified by the model (the higher the better), and compare models at different k values (ranging from 10 to 200).

To evaluate a model, we first compute the two metrics on the test set for each individual topic (i.e., industry) and then average across topics. It is important to note that the main focus of this paper is on estimating sharer reputation scores. Thus, improving model accuracy in terms of predicting individual users' response (e.g., click) to individual items is not the main goal. Because of this, we do not consider metrics that measure click prediction accuracy per event.

4.1 Comparison with Different Methods

In this subsection, we consider splitting data into a training set and a test set by time. In particular, we use the data in May for training, and the next 3 months for testing. We estimate sharers' reputation scores using the training data, and then evaluate the estimated scores based on average LT CTR computed using data in the test period. Notice that our test set is much larger than the training set. This is done to reduce the variance of the evaluation metric.

Baseline Methods. Many recent methods for finding influential users in social networks are based on PageRank (e.g., [30, 25, 28]). Thus, we consider PageRank as the main baseline to compare to. We implemented two versions of PageRank [4] for computing sharer reputation scores: *PageRank-Rate* and *PageRank-Volume*. These two methods are based on the intuition that users who share good items would necessarily get more responses from their connections. Let $N_V(i, s)$ be the number of items that are shared by sharer s and viewed by her connection i, and let $N_R(i, s)$ be the number of items that are shared by sharer s and responded (clicked or liked or re-shared) by her connection i. For example, if sharer s shared 11 items and her connection i saw 10 of them and responded to 3, then we have $N_V(i,s) = 10$ and $N_R(i,s) = 3$. Each method defines the edge weight $w(i,s)$ between each (recipient i, sharer s) pair based on $N_V(i,s)$ and $N_R(i,s)$ in its own way, and then computes the PageRank scores [4] for all of the sharers.

- *PageRank-Volume* defines edge weight $w(i,s)$ as the volume of responses from recipient i to sharer s: $w(i,s) = N_R(i,s)$.
- *PageRank-Rate* defines edge weight $w(i,s)$ as the response rate from recipient i to sharer s: $w(i,s) = N_R(i,j)/N_V(i,j)$.

Both methods normalize the edge weights so that the sum of outgoing edge weights is one: $\bar{w}(i,s) = \frac{w(i,s)}{\sum_s w(i,s)}$. Each model computes the score $f(s)$ of sharer s by solving the following equation:

$$f(s) = (1-\delta)\sum_i \bar{w}(i,s)f(i) + \frac{\delta}{N}$$

where δ is a given constant and N is the number of users. We experimented with a number of δ values, but only report the performance of $\delta = 0.2$ following [25]. We note that other values of δ do not improve the performance. To estimate reputation scores for industry k, we take an induced subgraph of users who follow or belong to industry k. Then, the PageRank scores on the induced subgraph (either by *PageRank-Rate* or *PageRank-Volume*) provides reputation scores for that industry.

We note that our PageRank methods follow the same approach as [30, 28]. We also tried another version of PageRank proposed by Sáez-Trumper et al. [25] in our early experiments. However, its performance is worse than *PageRank-Rate* or *PageRank-Volume* and gives negative correlations probably because it tends to identify abusive users. So, we drop it from our experiments.

Variations of our model. To measure the effect of adding different components to our hierarchical model, we consider two simplified versions of Full Model: Model (only Y) and Model (only Z). Model (only Z) assumes that the reputation of sharer s on topic k is its uncalibrated score α_{sk} estimated based only on biased social response data. Specifically, Model (only Z) estimates α_{sk} using only the NUS response data z_{sij} (Equation 12) and then sets $\mu_{sk} = \alpha_{sk}$. Model (only Y) estimates μ_{sk} using only unbiased user action data (the LT data) y_{ij} without using the NUS social response data at all. Specifically, Model (only Y) first estimates p_{jk} by solving Equation 10 and then estimates μ_{sk} through Equation 11. Note that Model (only Y) cannot estimate the reputation scores of "cold-start" sharers, who do not share any items recommended by LinkedIn Today during the training period.

Comparing overall performance of different models. The first row of Table 2 shows the performance of each model. Our Full Model achieves the highest rank correlation, which indicates that appropriate combination of biased social response (NUS data Z) with unbiased user action (LT data Y) improves the accuracy. Model (only Y) achieves the second highest rank correlation and Model (only Z) is the third. This is expected since user reputation is primarily defined based on unbiased LT data. For sharers with a sufficient amount of unbiased LT data, the biased NUS data is not really needed. However, for sharers who do not have many shared items in LT data, NUS data helps. Note that the PageRank baselines perform poorly — they show slight negative rank correlations.

As discussed in Section 3, the main purpose of using biased social response data (NUS data Z) is to estimates the reputation of *cold-start* users who do not share any item (or share very few items) in the unbiased user action data (LT data Y). To measure the performance of the methods in the cold-start scenario, we limit our focus on "cold-start test sharers" who did not share any item in the unbiased LT data. To reduce the variance of the performance metric, we increase the number of the cold-start test sharers by randomly picking 50% of test sharers and removing them from the unbiased training data. The second row of Table 2 shows the performance of different methods for these cold-start test sharers. Recall that Model (only Y) cannot estimate reputation scores for cold-start sharers. Full Model outperforms all the other methods with a significant margin. It achieves rank correlation 0.1124, a 241% lift over the second best method, Model (only Z).

Figure 3 shows the CTR of top k sharers (relative to the CTR of all shared items) for different models as a function k for the cold-start scenario. First, note that all curves would eventually converge to 1.0 when k is the total number of sharers since, by definition, the CTR of all sharers equals the average CTR of all shared items. Second, note that our Full Model significantly and uniformly outperforms all other methods. Also note that the other three curves, which compute reputation purely on biased social response data, are all below 1.0 when k is small. This suggests that users who receive highest social response seem to exhibit some undesired behavior (e.g., LinkedIn Open Networkers whose goals are connecting to as many people as possible, forming a large densely connected component with a large number of activities).

Breaking down by industries. We examine how the performance of different models varies depending on the topics (industries). Due to space limitation, we only show the results of Kendall's τ. Figure 4 shows box plots for the distributions of Kendall's τ across industries for the all user scenario and the cold-start scenario. Note that this box plot does not represent statistical significance, but shows the variation of rank correlation across industries. In Figure 4a, Full Model achieves higher τ in most industries; the 75% quantile of Full Model is higher than the median τ of the best baseline (Model (only Y)). On the other hand, the PageRank-based baselines and Model (only Y) perform poorly regardless of industries. In Figure 4b, we see a larger gap in performance than the gap in Figure 4a. The minimum value of Full Model is higher than the median value of the best baseline (Model (only Z)).

Figure 5 shows finer breakdown of the performance of Full Model compared to the best baseline for the top 20 industries for the two scenarios, where the y-axis $\Delta\tau$ is the difference between τ of Full Model and τ of the best baseline, for each industry (positive values mean Full Model outperform), and the bar width is proportional to the number of test users in the industry. In Figure 5a, Full Model outperforms in 14 (70% of all) industries and more importantly, the industries where Full Model cannot outperform have smaller numbers of sharers — the average number of test sharers in industries where Full Model is inferior is 56% of the number of the test users in the rest of the industries. In Figure 5b, Full Model is superior in most industries — it outperforms the best baseline method in 18 (90% of all) industries.

4.2 Model Characteristics

Different amounts of unbiased data. We first study the effect of using different amounts of unbiased user action data (the LT data) in model training. To control the amount of unbiased data, we create a series of datasets in the following way: Instead of splitting by time (as in Section 4.1), here we split by items. Specifically, we consider all of the items that appear in the unbiased LT data and randomly select 40% of the items to be the "test items" and call the rest 60% "training items". We set aside all of the events on

Model	*PageRank-Rate*	*PageRank-Volume*	Model (only Y)	Model (only Z)	Full Model
Kendall's τ (All)	-0.0176	-0.0215	0.1879	0.0211	0.2078
Kendall's τ (Cold-Start)	-0.0325	-0.0365	N/A	0.0329	0.1124

Table 2: Overall performance of different models in Kendall's τ rank correlation. "All" indicates the all users scenario. "Cold-Start" indicates the cold-start scenario, where test users do not have unbiased training sets. Simulation shows that the 95% confidence interval of a model that just assign random scores to users is between -0.011 and 0.011. To give a sense of an upper bound, we compute Kendall's τ between the ranking of users by their average LT CTRs (our proxy for ground-truth reputation scores) computed using the training data and that computed using the test data, for (user, topic) pairs with sufficient data (the user shared at least 10 articles on the topic). The upper bound is 0.3897.

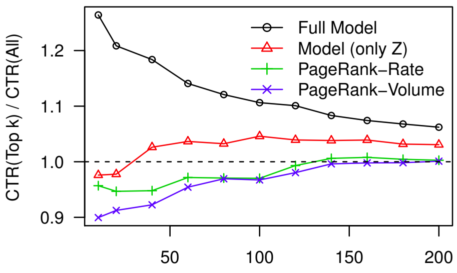
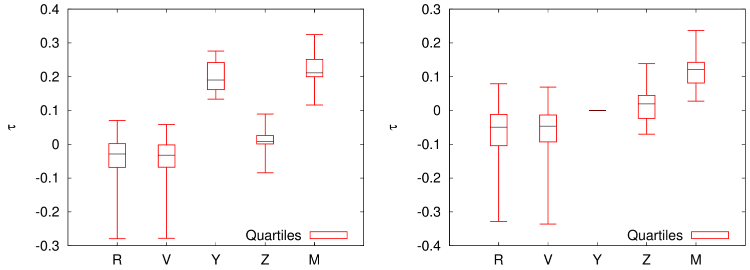


Figure 3: CTR of top k sharers for different models as a function of k, normalized by the average CTR of all items



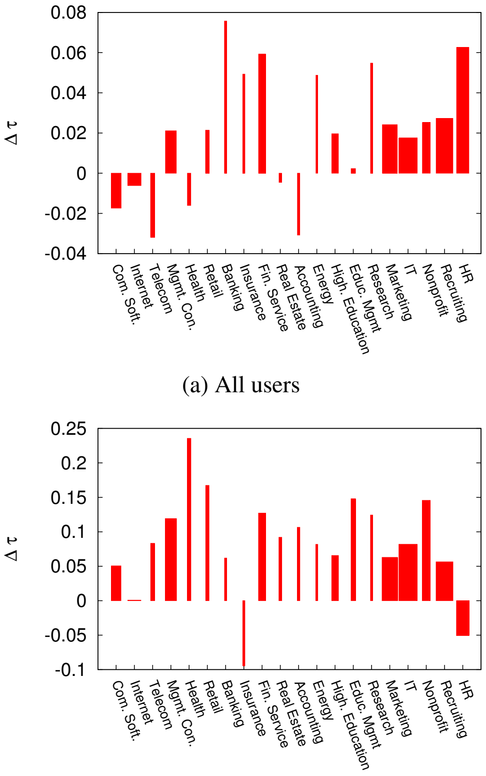
(a) All users

(b) Users with cold-start

Figure 4: Box plots of the distributions of Kendall's τ for the top 20 industries. R: *PageRank-Rate*, V: *PageRank-Volume*, Y: Model (only Y), Z: Model (only Z), M: Full Model.

the test items to form the test set. The remaining events form the "initial training set". Because our main focus is on how to estimate sharer reputation for sharers without unbiased data (cold-start), we randomly select 50% sharers who satisfy the two test sharer conditions (defined when we introduce evaluation metrics) to form the set of test sharers. We then remove all the events related to the test sharers from the initial training set to form the "100% training set", which consists of events on 16,545 items. To simulate different amounts of unbiased data, we randomly sample $P\%$ of the items that appear in the "100% training set" and select only training events on the sampled items to form a "$P\%$ training set". We note that model performance numbers in this subsection is not comparable to those in the previous subsection due to different ways of splitting data into training and test sets.

Figure 6a shows Kendall's τ rank correlation as a function of the percentage of unbiased training data used to train our model. As can be seen, the unbiased data can be reduced by 50% without significantly affecting model performance. From 50% to 5%, model performance degrades roughly in a log-linear manner (notice that x-axis is log scaled).



(a) All users

(b) Users with cold-start

Figure 5: Improvement in Kendall's τ for each industry by Full Model compared to the best baseline. The bar width is proportional to the number of test users in the industry

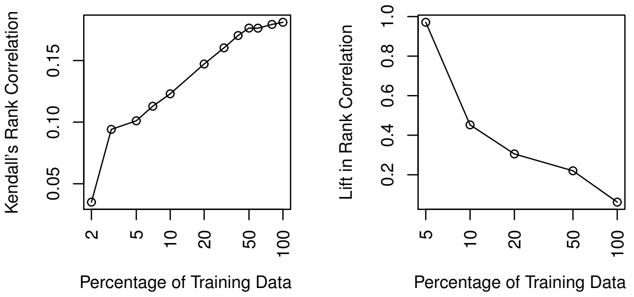
Co-sharing random field prior vs. zero-mean prior. Recall that in our model, we put a co-sharing random field prior over μ_{sk}. To understand the benefit of using co-sharing random field prior, we compare our model with this prior to the model that replaces the co-sharing random field prior with a commonly used alternative prior, a zero-mean prior; i.e.,

$$\mu_{sk} \sim \mathcal{N}(0, 1/\lambda_4). \tag{14}$$

Figure 6b shows the lift in Kendall's τ rank correlation of our model with the co-sharing random field prior over the model with the zero-mean prior as a function of the percentage of unbiased training data, where lift is defined as $(\tau_{\text{CSH-MRF}} - \tau_{\text{zero-mean}})/\tau_{\text{zero-mean}}$. As can be seen, the smaller the data size, the higher the lift.

4.3 Discussion

Our experiments provide interesting insights which we summarize below.



(a) Our model performance

(b) Lift over zero-mean

Figure 6: Effect of different amounts of unbiased data (x-axis is log scaled)

- Classical methods of estimating social influence like PageRank perform poorly. Our method that corrects bias in social response data with unbiased user action data through a novel multi-level hierarchical model provides significant improvement in performance.
- Our method of combining information from both sources is particularly effective in cold-start scenarios, *i.e.*, for predicting reputation of users who have not shared in the unbiased context. This is especially encouraging since we can use our method to increase coverage of reputation scores.
- It is interesting to see the role the co-sharing Markov random field prior plays in estimation. When the unbiased user action data is sparse, the additional smoothing provided by this prior leads to significant improvement in performance. This is good news since in practice, it is not always easy to obtain a large amount of data from an unbiased context.

5. CONCLUSION

We proposed a novel multi-level hierarchical model that estimates unbiased reputation scores by augmenting social response data that has selection and response bias with aggregated data from an unbiased context. Our method provides significant improvement over existing methods to estimate social influence. We show that it is indeed possible to correct for the biases in social response data by *benchmarking* it with reasonable amount of data from an unbiased context. Furthermore, the co-sharing Markov random field prior helps in providing reliable estimates even with small amount of unbiased data. Our research has opened up avenues for future work. While we illustrated our method with only two sources of data, one can generalize the method to multiple "misaligned" data sources; some of them may be unbiased but available at different levels of aggregation, while others could be more granular but biased. For instance, one can obtain aggregated unbiased data from different kinds of recommendation modules. It is even possible to obtain some measures of article quality through offsite reading and sharing data obtained from external publishers that are instrumented with "social share buttons". Developing a principled framework to combine multiple misaligned social response data available at different levels of aggregations with different kinds of biases is a challenging problem we plan to address in the future.

6. REFERENCES

[1] D. Agarwal, B.-C. Chen, and P. Elango. Explore/exploit schemes for web content optimization. In *ICDM*, 2009.

[2] A. Anderson, D. P. Huttenlocher, J. M. Kleinberg, and J. Leskovec. Discovering value from community activity on focused question answering sites: a case study of stack overflow. In *KDD*, 2012.

[3] A. Anderson, D. P. Huttenlocher, J. M. Kleinberg, and J. Leskovec. Effects of user similarity in social media. In *WSDM*, 2012.

[4] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks*, 1998.

[5] M. Cha, H. Haddadi, F. Benevenuto, and P. K. Gummadi. Measuring user influence in twitter: The million follower fallacy. In *ICWSM*, 2010.

[6] B.-C. Chen, A. Dasgupta, X. Wang, and J. Yang. Vote calibration in community question-answering systems. In *SIGIR*, 2012.

[7] B.-C. Chen, J. Guo, B. L. Tseng, and J. Yang. User reputation in a comment rating environment. In *KDD*, 2011.

[8] D. J. Crandall, D. Cosley, D. P. Huttenlocher, J. M. Kleinberg, and S. Suri. Feedback effects between similarity and social influence in online communities. In *KDD*, 2008.

[9] C. Danescu-Niculescu-Mizil, M. Gamon, and S. T. Dumais. Mark my words!: linguistic style accommodation in social media. In *WWW*, 2011.

[10] K. El-Arini, U. Paquet, R. Herbrich, J. V. Gael, and B. A. y Arcas. Transparent user models for personalization. In *KDD*, 2012.

[11] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. Liblinear: A library for large linear classification. *JMLR*, 2008.

[12] W.-S. Hwang, H.-J. Lee, S.-W. Kim, and M. Lee. On using category experts for improving the performance and accuracy in recommender systems. In *CIKM*, 2012.

[13] U. Kang, S. Papadimitriou, J. Sun, and H. Tong. Centralities in large networks: Algorithms and observations. In *SDM*, 2011.

[14] E. Katz and P. Lazarsfeld. *Personal influence: the part played by people in the flow of mass communications*. Foundations of communications research. 1955.

[15] D. Kempe, J. M. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *KDD*, 2003.

[16] Y. Kim and K. Shim. Twitobi: A recommendation system for twitter using probabilistic modeling. In *ICDM*, 2011.

[17] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *JACM*, 1999.

[18] P. Li, J. X. Yu, H. Liu, J. He, and X. Du. Ranking individuals and groups by influence propagation. In *PAKDD*, 2011.

[19] B. Liu and L. Zhang. A survey of opinion mining and sentiment analysis. In *Mining Text Data*. 2012.

[20] L. Liu, J. Tang, J. Han, M. Jiang, and S. Yang. Mining topic-level influence in heterogeneous networks. In *CIKM*, 2010.

[21] A. S. Maiya and T. Y. Berger-Wolf. Online sampling of high centrality individuals in social networks. In *PAKDD*, 2010.

[22] A. Pal and S. Counts. Identifying topical authorities in microblogs. In *WSDM*, 2011.

[23] B. A. Prakash and C. Faloutsos. Understanding and managing cascades on large graphs. *PVLDB*, 2012.

[24] M. Sachan, D. Contractor, T. A. Faruquie, and L. V. Subramaniam. Using content and interactions for discovering communities in social networks. In *WWW*, 2012.

[25] D. Sáez-Trumper, G. Comarela, V. A. F. Almeida, R. A. Baeza-Yates, and F. Benevenuto. Finding trendsetters in information networks. In *KDD*, 2012.

[26] T. Sakai, D. Ishikawa, N. Kando, Y. Seki, K. Kuriyama, and C.-Y. Lin. Using graded-relevance metrics for evaluating community qa answer selection. In *WSDM*, 2011.

[27] Y. R. Tausczik and J. W. Pennebaker. Participation in an online mathematics community: differentiating motivations to add. In *CSCW*, 2012.

[28] G. Wang, Y. Zhao, X. Shi, and P. S. Yu. Magnet community identification on social networks. In *KDD*, 2012.

[29] D. J. Watts and P. S. Dodds. Influentials, networks, and public opinion formation. *Journal of Consumer Research*, 2007.

[30] J. Weng, E.-P. Lim, J. Jiang, and Q. He. Twitterrank: finding topic-sensitive influential twitterers. In *WSDM*, 2010.