Information Sciences 302 (2015) 70–82

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Detecting high-quality posts in community question answering sites

Yuan Yao [a], Hanghang Tong [b], Tao Xie [c], Leman Akoglu [d], Feng Xu [a,*], Jian Lu [a]

[a] State Key Laboratory for Novel Software Technology, Nanjing University, China
[b] Arizona State University, USA
[c] University of Illinois at Urbana-Champaign, USA
[d] Stony Brook University, USA

ARTICLE INFO

Article history:
Received 3 August 2014
Received in revised form 15 November 2014
Accepted 9 December 2014
Available online 12 January 2015

Keywords:
CQA
Question and answer
Voting correlation
Voting prediction

ABSTRACT

Community question answering (CQA) has become a new paradigm for seeking and sharing information. In CQA sites, users can ask and answer questions, and provide feedback (e.g., by voting or commenting) to these questions/answers. In this article, we propose the early detection of high-quality CQA questions/answers. Such detection can help discover a high-impact question that would be widely recognized by the users in these CQA sites, as well as identify a useful answer that would gain much positive feedback from site users. In particular, we view the post quality from the perspective of the voting outcome. First, our key intuition is that the voting score of an answer is strongly positively correlated with that of its question, and we verify such correlation in two real CQA data sets. Second, armed with the verified correlation, we propose a family of algorithms to jointly detecting the high-quality questions and answers soon after they are posted in the CQA sites. We conduct extensive experimental evaluations to demonstrate the effectiveness and efficiency of our approaches. Overall, our algorithms can outperform the best competitor in prediction performance, while enjoying linear scalability with respect to the total number of posts.

1. Introduction

Community question answering (CQA) has become a new paradigm for seeking and sharing information. For example, millions of users now use CQA sites to search for solutions for their problems [15,19]. Example CQA sites include those general ones such as Yahoo! Answers[1] and Baidu Knows,[2] and those domain-specific ones like Stack Overflow[3] and Mathematics Stack Exchange.[4]

One major difference between CQA and traditional QA is from the volunteer efforts of site users. In addition to posting questions/answers, most of the existing CQA sites allow the site users to vote (e.g., upvote and downvote in Stack Overflow)

* Corresponding author.
E-mail addresses: yyao@smail.nju.edu.cn (Y. Yao), hanghang.tong@asu.edu (H. Tong), taoxie@illinois.edu (T. Xie), leman@cs.stonybrook.edu (L. Akoglu), xf@nju.edu.cn (F. Xu), lj@nju.edu.cn (J. Lu).

[1] http://answers.yahoo.com/.
[2] http://zhidao.baidu.com/.
[3] http://stackoverflow.com/.
[4] http://math.stackexchange.com/.

http://dx.doi.org/10.1016/j.ins.2014.12.038
0020-0255/

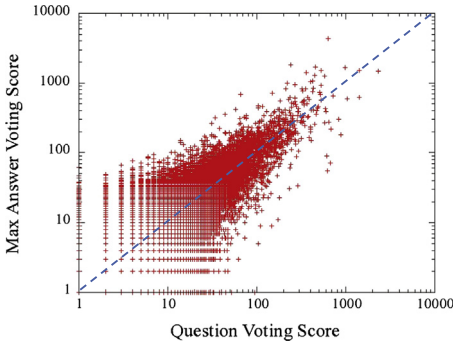


Fig. 1. The strong voting correlation between questions and their best answers in *SO* dataset. Pearson correlation coefficient $r = 0.6665$ with p-value < 0.0001. See Fig. 2 for more results.

for these questions/answers. On one hand, the voting mechanism as well as its reputation system provides the main incentives for tight involvement and productive competition of the whole community. On the other hand, the outcome of such voting, e.g., the difference between the number of the upvotes and downvotes that a question/answer receives from the site users (referred to as 'voting score'), provides a good indicator of the intrinsic value of a question/answer. To some extent, the voting score of a question/answer resembles the number of the citations that a research paper receives in the scientific publication domain. It reflects the net number of users who have a positive attitude toward the paper.

In this article, we view the post quality from the perspective of the voting outcome, and propose the early detection of high-quality CQA questions/answers. To date, a lot of efforts have been made to study the quality prediction problem in CQA sites. However, most of them treat questions and answers separately (see Section 6 for a review).

We conjecture that there exists *correlation* between the voting score of a question and that of its associated answer. Intuitively, an interesting question might obtain more attention from potential answerers and thus has a better chance to receive high-score answers. On the other hand, it might be very difficult for a low-score question to attract a high-score answer due to, e.g., its poor expression in language, or lack of interestingness in topic. Starting from this conjecture, we study two real CQA sites, i.e., Stack Overflow (*SO*), and Mathematics Stack Exchange (*Math*). Our key finding is that the voting score of an answer is indeed strongly positively correlated with that of its question (see Fig. 1). Such correlation structure consistently exists on both sites.

Armed with this observation, we propose a family of co-prediction algorithms (*CoPs*) to *jointly* predict the voting scores of questions and answers. In particular, we aim at identifying the potentially high-score posts soon after they are posted in the CQA sites. We conduct extensive experimental evaluations to demonstrate the effectiveness and efficiency of our approaches. Overall, our *joint* prediction approaches achieve up to 15.2% net precision improvement for answer prediction over the best competitor in one of the data sets we studied. In addition, the proposed *CoPs* algorithms enable us to predict the voting outcome of an answer *before* it actually appears on the site. Finally, our approaches scale linearly wrt the total number of questions and answers.

The main contributions of this paper include:

- *Empirical Findings*. We empirically study the voting scores of posts in CQA sites. To the best of our knowledge, we are the first to *quantitatively* validate the correlation between the voting scores of questions and those of their associated answers on two independent real data sets.
- *Algorithms and Evaluations*. We propose a family of co-prediction algorithms (*CoPs*) to jointly predict the voting scores of questions and answers. We further perform extensive experimental evaluations on two real data sets to demonstrate the effectiveness and efficiency of our approaches.

The rest of the paper is organized as follows. Section 2 presents the empirical studies about the voting scores of questions/answers. Sections 3 and 4 present the problem definitions and the proposed algorithms for the joint voting prediction problem, respectively. Section 5 presents the experimental results. Section 6 reviews related work, and Section 7 concludes the paper.

2. Empirical study

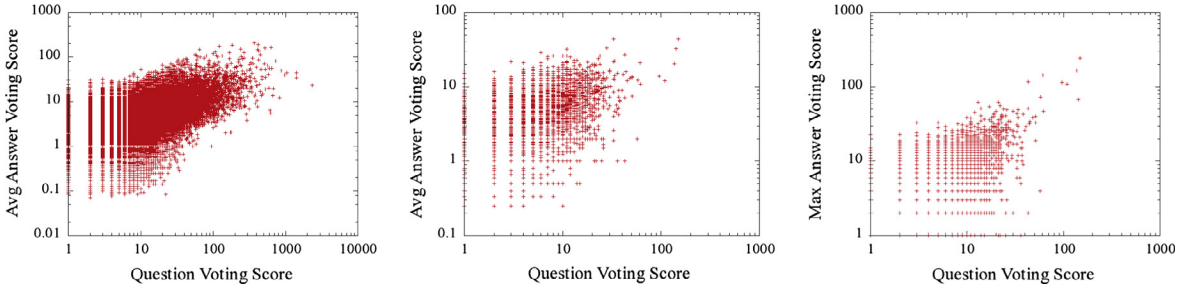
In this section, we perform an empirical study of the voting scores of questions and answers in *SO* (Stack Overflow) and *Math* (Mathematics Stack Exchange) data sets. They are popular CQA sites for programming and math, respectively. For both data sets, they are officially published and publicly available.[5] The statistics of the two data sets are summarized in Table 1.

We first study the overall correlation between the voting scores of questions and those of their answers. For a given question, there might be multiple answers. Thus, we report both the highest (i.e., the best answer) and the average voting scores

[5] http://blog.stackoverflow.com/category/cc-wiki-dump/.

Table 1
The statistics of *SO* and *Math* datasets.

Data	Questions	Answers	Users	Votes
SO	1,966,272	4,282,570	756,695	14,056,000
Math	16,638	32,876	12,526	202,932



(a) Average answer score vs. Question score in *SO* data ($r = 0.5424$, p-value < 0.0001)

(b) Average answer score vs. Question score in *Math* data ($r = 0.5022$, p-value < 0.0001)

(c) Maximum answer score vs. Question score in *Math* data ($r = 0.6543$, p-value < 0.0001)

Fig. 2. The strong voting correlation between questions and their answers in *SO* and *Math*. r stands for the Pearson correlation coefficient. The maximum answer score vs. question score for *SO* data is in Fig. 1.

of its answers. The results are shown in Figs. 1 and 2, where the Pearson correlation coefficient r is also computed. As we can see from the figures, the scores of questions and those of their answers are strongly correlated in both data sets.

Next, we study the voting correlation between questions and their answers over time. Here, we compute the correlation between the scores of questions and the average scores of their answers, and show the result over several time snapshots on *SO* data in Fig. 3. As we can see, the strong positive voting correlation consistently exists across all the time snapshots – even at the very early stage (e.g., $r > 0.5$). This result indicates that it is doable to employ the early voting correlation to predict the future voting scores of questions/answers.

To gain a deeper understanding of the voting correlation, we further divide questions and answers into several bins based on their scores. For example, questions and answers in *SO* are divided into 11 bins (i.e., $Q1, Q2, \ldots, Q11$, and $A1, A2, \ldots, A11$) where the score s in each bin is in the range of $s < 0, s = 0, s = 1, s = 2, s = 3, s = 4, s = 5, 6 \leqslant s \leqslant 10, 11 \leqslant s \leqslant 50, 51 \leqslant s \leqslant 100, s > 100$, respectively. Based on the divided bins, we next study the distribution of answers over each question bin. That is, for questions in bin Qi, we show the percentage of associated answers over the 11 answer bins in Fig. 4.

Several interesting observations can be found from the figures. First, we can see that the heat-maps of the two data sets exhibit a very similar pattern, despite the fact that they are about two totally different subjects (programming language vs. math). Second (Zone-1), the majority of questions and answers are in the vicinity of the diagonal, which indicates the positive correlation between question scores and answer scores. Third (Zone-2), the color in the bottom-right corner is very dark, which verifies that it is rare to have high-score answers when the question is of low score. Fourth (Zone-3), in the top-left corner, some high-score questions may have low-score answers. This is probably due to the implicit competition among different answers, i.e., one of the multiple answers for a question attracts most of the community attention (e.g., 96.6% of the questions in this zone have at least one answer whose score is larger than 5 in *SO* data). Finally (Zone-4), most high-score answers are in the top-right corner of the figure, indicating that their questions are also of high score.

Based on these results, we conclude that a strong positive correlation of voting scores between questions and their answers does exist in both *SO* and *Math* data sets.

3. Joint voting prediction: problem definitions

In this section, we present the problem definition for voting prediction of questions/answers in CQA sites.

Let us first introduce the notations that we use throughout the paper in Table 2. Following conventions, we use bold capital letters for matrices, and bold lower case letters for vectors. For example, we use $\mathbf{X}_q$ to denote the feature matrix for questions where each row contains the feature vector for the corresponding question. In this work, we adopt some commonly used features for CQA sites such as the number of comments and the length of the post (See the experiment section for details on feature extraction). We use the $n_q \times n_a$ matrix $\mathbf{M}$ to denote the association matrix of questions and answers where $\mathbf{M}(i,j) = 1$ indicates that the jth answer belongs to the ith question. Similar to Matlab, we also denote the ith row of matrix $\mathbf{M}$ as $\mathbf{M}(i,:)$, and the transpose of a matrix with a prime (i.e., $\mathbf{M}' \equiv \mathbf{M}^T$).

Based on the above notations, the prediction problem for questions could be defined as:

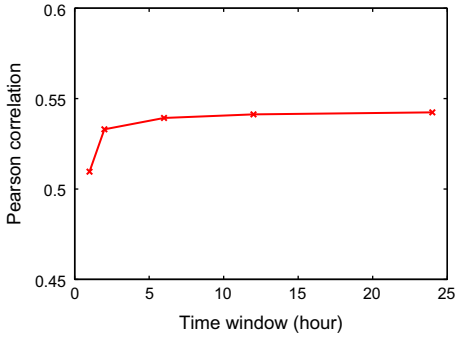
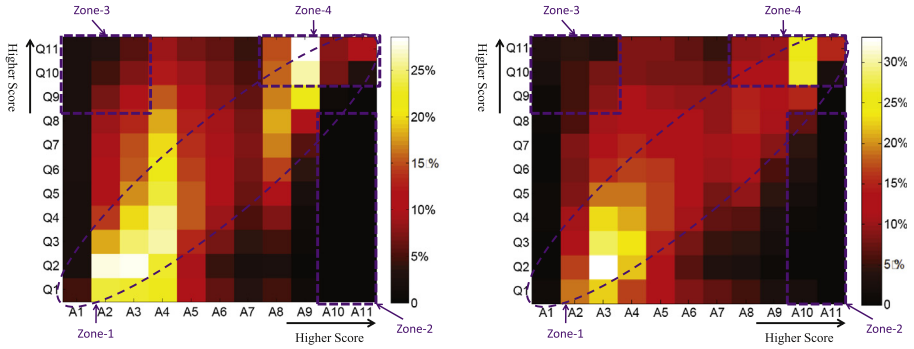


Fig. 3. The voting correlation between questions and their answers over time. The *y*-axis represents the Pearson correlation coefficient *r*, and the *x*-axis represents the time after the question is posted. For all *r*, *p*-value < 0.0001.



(a) *SO* data (b) *Math* data

Fig. 4. The heat-maps of voting score distribution over the divided bins (best view in color). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 2
Symbols.

Symbol	Definition and description
$\mathbf{X}_q, \mathbf{X}_a$	The feature matrix for questions and answers
$\mathbf{X}_{\bar{q}}, \mathbf{X}_{\bar{a}}$	The feature matrix with transferred features
$\mathbf{y}_q, \mathbf{y}_a$	The final voting score of questions and answers
$\boldsymbol{\beta}_q, \boldsymbol{\beta}_a$	The coefficients for the features
$\mathbf{M}$	The association matrix of questions and answers
$\bar{\mathbf{M}}$	The row-normalized matrix of $\mathbf{M}$
$\mathbf{M}'$	The transpose of matrix $\mathbf{M}$
$\mathbf{M}(i,j)$	The element at the ith row and jth column of $\mathbf{M}$
$\mathbf{M}(i,:)$	The ith row of matrix $\mathbf{M}$
n_q, n_a	The number of questions and answers
m	The maximum iteration number
ξ	The threshold to terminate the iteration

Problem 1 (*The voting prediction of questions*).

Given: the question feature matrix $\mathbf{X}_q$, and the vector of question voting scores $\mathbf{y}_q$;
Find: the final voting score of a new question.

An analogous definition for answer voting prediction can be defined, and we omit the definition for brevity. For a given question/answer, its final voting score is defined as the difference between the number of the upvotes and downvotes that a question/answer receives from the site users. Yet, the individual upvote/downvote could span a long period. For instance, some questions/answers might still receive upvotes/downvotes one year after they were posted on the site. Our goal is to

predict the final voting scores of questions/answers in a short period after they were posted. Therefore, we can only use the available information in this short period. For example, if we need to predict the final voting in one hour, the feature matrices $\mathbf{X}_q$ and $\mathbf{X}_a$ should contain only the information that is available in the first hour after the question is posted.

One of our key findings in the previous section is the strong positive voting correlation between questions and their answers. To leverage such correlation, we take the association matrix $\mathbf{M}$ into account, and jointly predict the voting scores of questions/answers as:

Problem 2 (*The joint voting prediction of question and answer*).

Given: the question feature matrix $\mathbf{X}_q$, the answer feature matrix $\mathbf{X}_a$, the vector of question voting scores $\mathbf{y}_q$, the vector of answer voting scores $\mathbf{y}_a$, and the association matrix $\mathbf{M}$;
Find: the final voting scores of a new question and its answers.

4. Joint voting prediction: proposed approaches

In this section, we present our algorithms to solve Problem 2. That is, we want to jointly predict the *final* voting scores for questions and answers. After we introduce some preliminaries, we start with the intuitions and basic strategies behind our algorithms. Then, we present and analyze our co-prediction algorithms (*CoPs*).

4.1. Preliminaries: separate methods

Here, we first introduce the separate method that could be used for the question voting prediction in Problem 1. Similar method can be applied for answer prediction. Specially, the problem can be formulated as the optimization problem:

$$\mathcal{L}_0 = \min_{\boldsymbol{\beta}_q} \underbrace{\sum_{i=1}^{n_q} g(\mathbf{X}_q(i,:)\boldsymbol{\beta}_q, \mathbf{y}_q(i))}_{\text{question prediction}} + \underbrace{\lambda||\boldsymbol{\beta}_q||_2^2}_{\text{regularization}} \tag{1}$$

where g indicates the loss function, and $\lambda||\boldsymbol{\beta}_q||_2^2$ is used to avoid over-fitting. In this formulation, various loss functions could be used. For example, if we choose square loss for g, the problem becomes the standard ridge regression problem. Then, the formulation can be solved by the closed-form solution: $\boldsymbol{\beta}_q = (\mathbf{X}_q'\mathbf{X}_q + \lambda\mathbf{I})^{-1}\mathbf{X}_q'\mathbf{y}_q$. Once the coefficient vector $\boldsymbol{\beta}_q$ is learnt, it can be used to predict the voting score for a new question. For example, in the above case, the predicted voting score for a new question is simply the inner product between its feature vector and the coefficient vector $\boldsymbol{\beta}_q$.

4.2. Intuitions and basic strategies

Next, we present the basic strategies that we explore to leverage the observed voting correlation. There are two basic strategies behind our *CoPs* algorithms. We summarize them together with the intuitions behind each strategy as follows.

S1 Feature expansion: The first strategy considers to expand the feature space. Because the scores of questions and those of their answers are correlated, the features for question prediction are potentially useful for answer prediction. As a result, we transfer the question features to $\mathbf{M}'\mathbf{X}_q$ and add these features for answer voting prediction. Namely, we use $\mathbf{X}_{\bar{a}} = [\mathbf{X}_a, \mathbf{M}'\mathbf{X}_q]$ to represent the new feature matrix for answers. Similarly, we transfer the answer features to $\tilde{\mathbf{M}}\mathbf{X}_a$ and incorporate these features with $\mathbf{X}_q$ as $\mathbf{X}_{\bar{q}} = [\mathbf{X}_q, \tilde{\mathbf{M}}\mathbf{X}_a]$. We use the row-normalized $\tilde{\mathbf{M}}$ matrix in the latter case. In other words, for a question with multiple answers, we take the average of the features from these answers.

S2 Voting consistency: The second strategy takes into account the consistency in the label space. That is, for a pair of question and answer, we could directly maximize the voting correlation or minimize the voting difference between them. In this work, we try to minimize the difference between the predicted score of a question and that of its answer. For instance, we can require that $\hat{\mathbf{y}}_q \approx \tilde{\mathbf{M}}\hat{\mathbf{y}}_a$, where we constrain that the predicted question score is close to the predicted average score of its answers.

4.3. The general procedure

Based on the above two strategies (i.e., feature expansion and voting consistency), we propose a new optimization formulation for the joint voting prediction problem (i.e., Problem 2). That is, after transferring the features, we add an additional voting consistency constraint into the following optimization formulation:

$$\mathcal{L} = \min_{\boldsymbol{\beta}_q, \boldsymbol{\beta}_a} \underbrace{\frac{1}{n_q}\sum_{i=1}^{n_q} g(\mathbf{X}_{\bar{q}}(i,:)\boldsymbol{\beta}_q, \mathbf{y}_q(i))}_{\text{question prediction}} + \underbrace{\frac{1}{n_a}\sum_{i=1}^{n_a} g(\mathbf{X}_{\bar{a}}(i,:)\boldsymbol{\beta}_a, \mathbf{y}_a(i))}_{\text{answer prediction}} + \underbrace{\frac{\eta}{n_q}\sum_{i=1}^{n_q} h(\mathbf{X}_{\bar{q}}(i,:)\boldsymbol{\beta}_q, \tilde{\mathbf{M}}(i,:)\mathbf{X}_{\bar{a}}\boldsymbol{\beta}_a)}_{\text{voting consistency}} + \underbrace{\lambda(||\boldsymbol{\beta}_q||_2^2 + ||\boldsymbol{\beta}_a||_2^2)}_{\text{regularization}} \tag{2}$$

where h indicates the loss function of the additional voting consistency term, and η is a parameter to control the importance of this term. We also normalize the three terms (question prediction, answer prediction, and voting consistency) in the objective function so that the contribution of each question/answer is balanced.

The optimization framework in Eq. (2) is pretty general and many loss functions for g and h can be plugged in. In this work, we consider square loss, sigmoid loss, and logistic loss. Three loss functions are shown in Eq. (3).

$$\begin{aligned} g_{square}(x,y) &= (x-y)^2 \\ g_{logistic}(x,y) &= -y\log\frac{1}{1+\exp(-x)} - (1-y)\log(1-\frac{1}{1+\exp(-x)}) \\ g_{sigmoid}(x,y) &= \frac{1}{1+\exp(xy)} \end{aligned} \tag{3}$$

The rationality of these loss functions is as follows. Since our goal is to identify high-score posts, the difference between the real voting score and the estimated voting score (square loss), and the consistency between the real voting label and the estimated label (logistic loss and sigmoid loss) are both important for our task. To be specific, we divide the question/answer posts into two classes: high-score posts (labeled as +1) and low-score posts (labeled as 0 for logistic loss, and −1 for square loss and sigmoid loss).

We have five variants from Eq. (2) by setting g and h based on the three loss functions, as shown in Table 3.

Algorithm 1. Algorithm skeleton for *CoPs*

Input: $\mathbf{X}_q, \mathbf{X}_a, \mathbf{y}_q, \mathbf{y}_a$, and $\mathbf{M}$
Output: $\boldsymbol{\beta}_q$ and $\boldsymbol{\beta}_a$
1: $\mathbf{X}_{\tilde{q}} \leftarrow [\mathbf{X}_q, \tilde{\mathbf{M}}\mathbf{X}_a]$;
2: $\mathbf{X}_{\tilde{a}} \leftarrow [\mathbf{X}_a, \mathbf{M}'\mathbf{X}_q]$;
3: initialize $\boldsymbol{\beta}_q$ and $\boldsymbol{\beta}_a$;
4: **while** not convergent **do**
5: $\quad \boldsymbol{\beta}_q \leftarrow \boldsymbol{\beta}_q - \gamma \frac{\partial \mathcal{L}}{\partial \boldsymbol{\beta}_q}$ by Eq. (2);
6: $\quad \boldsymbol{\beta}_a \leftarrow \boldsymbol{\beta}_a - \gamma \frac{\partial \mathcal{L}}{\partial \boldsymbol{\beta}_a}$ by Eq. (2);
7: **end while**
8: **return** $\boldsymbol{\beta}_q$ and $\boldsymbol{\beta}_a$;

In order to solve the optimization problem in Eq. (2), our key observation is that for each of five cases in Table 3, the gradient for each term in Eq. (2) exists. This naturally leads to a gradient-descent type of iterative procedure to solve Eq. (2), as shown in Algorithm 1. As we can see, after initializing $\boldsymbol{\beta}_q$ and $\boldsymbol{\beta}_a$, we alternatively update these two coefficient vectors iteratively. We stop the iteration when the L_2 norm between successive estimates of both $\boldsymbol{\beta}_q$ and $\boldsymbol{\beta}_a$ is below the threshold ξ, or when the maximum iteration number m is reached. For steps 5–6, we can adopt either the batch gradient descent method with a learning step γ, or the Newton's method by substituting γ with the Hessian matrix of Eq. (2). The details of steps 5–6 with various loss functions are presented in the appendix for completeness.

4.4. A special case: CoPs-QQ

For the optimization problem in Eq. (2), if we set both g and h as square loss, we can have the closed-form solution for *CoPs-QQ*:

$$\begin{pmatrix} \boldsymbol{\beta}_q \\ \boldsymbol{\beta}_a \end{pmatrix} = \begin{pmatrix} \frac{\eta+1}{n_q}\mathbf{X}'_{\tilde{q}}\mathbf{X}_{\tilde{q}} + \lambda\mathbf{I} & -\frac{\eta}{n_q}\mathbf{X}'_{\tilde{q}}\tilde{\mathbf{M}}\mathbf{X}_{\tilde{a}} \\ -\frac{\eta}{n_q}\mathbf{X}'_{\tilde{a}}\tilde{\mathbf{M}}'\mathbf{X}_{\tilde{q}} & \frac{1}{n_a}\mathbf{X}'_{\tilde{a}}\mathbf{X}_{\tilde{a}} + \frac{\eta}{n_q}\mathbf{X}'_{\tilde{a}}\tilde{\mathbf{M}}'\tilde{\mathbf{M}}\mathbf{X}_{\tilde{a}} + \lambda\mathbf{I} \end{pmatrix}^{-1} \begin{pmatrix} \frac{1}{n_q}\mathbf{X}'_{\tilde{q}}\mathbf{y}_q \\ \frac{1}{n_a}\mathbf{X}'_{\tilde{a}}\mathbf{y}_a \end{pmatrix} \tag{4}$$

Table 3
The five variants derived from Eq. (2).

Algorithm	g	h
CoPs-QQ	Square loss	Square loss
CoPs-QG	Square loss	Sigmoid loss
CoPs-GG	Sigmoid loss	Sigmoid loss
CoPs-GQ	Sigmoid loss	Square loss
CoPs-LQ	Logistic loss	Square loss

4.5. Algorithm analysis

Here, we briefly analyze the effectiveness and efficiency of our algorithms for the optimization problem in Eq. (2).

The effectiveness of the proposed algorithms can be summarized in Lemma 1, which states that both Algorithm 1 and Eq. (4) can find a local minimum solution.

Lemma 1 (Effectiveness of CoPs). *Algorithm 1 finds a local minimum for the optimization problem in Eq. (2). In the special case, Eq. (4) also finds a local minimum.*

Proof. See the appendix. □

The time complexity and space complexity of the proposed *CoPs* are summarized in Lemmas 2 and 3, respectively. Basically, the lemmas state that *CoPs* scales linearly wrt the total number of questions and answers in both time and space.

Lemma 2 (Time efficiency of CoPs). *By storing matrix* $\tilde{\mathbf{M}}$ *in sparse matrix format, Algorithm 1 requires* $O((n_q + n_a)dm)$ *time with gradient descend method and* $O((n_q + n_a)dm + d^3m)$ *time with Newton's method, where d is the total number of features for questions and answers, and m is the maximum iteration number. Specifically, the algorithm in* Eq. (4) *requires* $O((n_q + n_a)d^2 + d^3)$ *time.*

Proof. See the appendix. □

Lemma 3 (Space efficiency of CoPs). *By storing matrix* $\tilde{\mathbf{M}}$ *in sparse matrix format, Algorithm 1 requires* $O((n_q + n_a)d)$ *space with gradient descend method and* $O((n_q + n_a)d + d^2)$ *space with Newton's method, where d is the total number of features for questions and answers. Specifically, the algorithm in* Eq. (4) *requires* $O((n_q + n_a)d + d^2)$ *space.*

Proof. See the appendix. □

5. Experiments

In this section, we present the experimental evaluations. The experiments are designed to answer the following questions:

- *Effectiveness*: How accurate are the proposed approaches for identifying the potentially high-score questions/answers?
- *Efficiency*: How do the proposed approaches scale?

5.1. Experimental setup

Here, our primary goal is to evaluate to what extent the voting correlation between questions and their answers could improve the prediction performance. We adopt some commonly used features in the literature, which are summarized in Table 4. For example, we use questioners'/answerers' reputation because it could reflect their ability of posting/answering a high-score question; we use the number of previous questions/answers because it relates to the experiences of the corresponding questioner/answerer; we also use the length of question/answer body and that of question title. For these features, we can extract them at the moment when the question/answer is posted. For others, we need to choose a short time window by the end of which the voting score is predicted. For example, we use the number of answers received in the time window for question voting prediction, and the number of comments received in the time window for both question and answer prediction. In this work, we fix the time window as one hour. Notice that in this paper, we have deliberately focused on these contextual features and bypassed the content of questions/answers. This is mainly due to the following two reasons. First, as we will show in the next subsection, these contextual features already give a reasonably high precision (e.g., near 88% precision for identifying high-score questions on *SO* data). Second, according to the complexity analysis (Lemma 2), both the existing and the proposed prediction algorithms could be *cubic* in the dimensionality of the features (e.g., the close-form solution for *CoPs-QQ* and the separate linear regression, Algorithm 1 and separate logistic regression with Newton's method, etc). Thus, bypassing the content-based features also helps to keep the algorithms scalable.

We formulate the task of quality prediction as a binary classification task, where we want to identify the small amount of high-quality questions/answers. As mentioned in introduction, we use the voting score of questions and answers as the quality indicator. When such voting results are not available, we can use other metrics such as question utility [25] and long-lasting value [3] as the quality indicator; or we may even require some human annotators to manually label the quality of posts (see Section 6 for more details). In this work, we define the posts whose score is no less than 10 as high-score posts

Table 4
Selected features for the voting prediction of questions/answers.

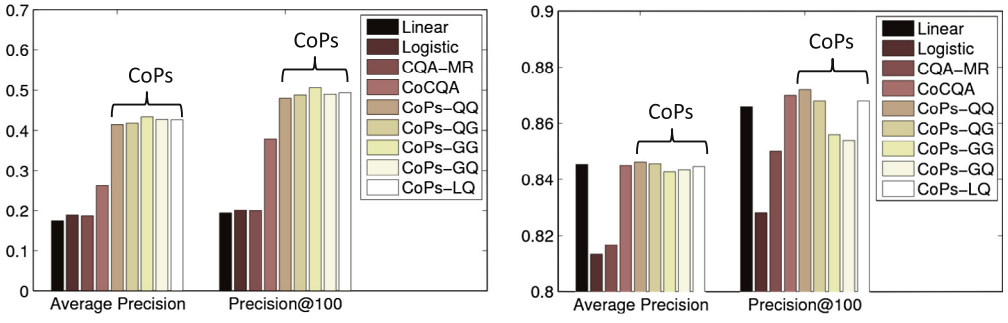
Feature description for questions
Questioner's reputation at question creation time
of questioner's previous questions at question creation time
The length of the question body
The length of the question title
of answers received in one hour after question creation
of comments received in one hour after question creation
Feature description for answers
Answerer's reputation at answer creation time
of answerer's previous answers at answer creation time
The length of the question answer
of comments received in one hour after question creation

in both *SO* and *Math* data sets. This results in 3.4% and 8.7% high-score posts in *SO* and *Math*, respectively. In other words, both data sets are highly skewed in terms of high-score posts vs. low-score ones. We report the precision of successfully identified high-score posts as the evaluation metric. For the readers who are interested in other evaluation metrics (e.g., classification accuracy in a balanced setting), please refer to our tech-report [31]. For each data set, we randomly choose 10% questions and their associated answers as the training set, and use the rest as the test set. Unless otherwise stated, for all the results reported here, we fix $m = 2000, \xi = 10^{-6}$, and $\gamma = 0.1$. For the two parameters η and λ in our methods, we experimentally found that our methods are robust with these two parameters in a large range. For the results that we report in this paper, we fix $\eta = 0.1$ and $\lambda = 10^{-4}$.

5.2. Effectiveness results

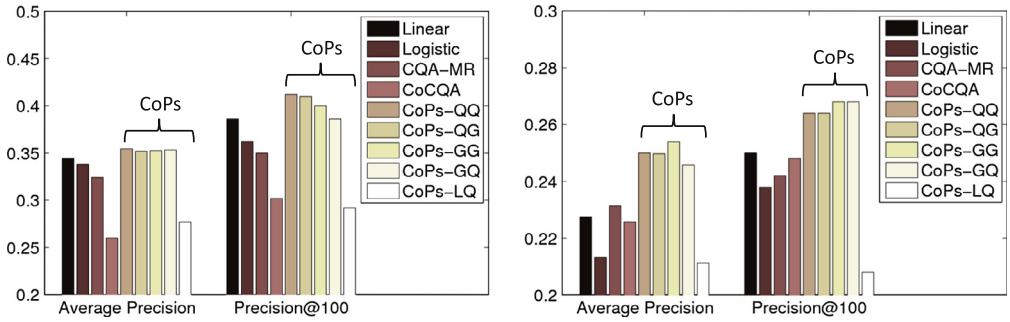
(A) The Effectiveness Comparisons of CoPs. For effectiveness, we first compare our methods with several existing methods, including the separate *Linear* regression method, the separate *Logistic* regression method, the *CQA-MR* method [5], and the *CoCQA* method [13]. The results on *SO* and *Math* are shown in Figs. 5 and 6, respectively. In the figures, we report the precision at 100 as well as the average precision over 10, 50, 100, 150, ... , 400. Since we randomly select the training data, we repeat the random selection 5 times and report the average results of these 5 experiments.

As we can see, overall, our *CoPs* methods outperform all the compared methods on both data sets. For example, the average precision of answer prediction by *CoPs-QQ* is 15.2% and 1.0% higher than the best competitor on *SO* and *Math*, respectively. For question prediction, all the methods can achieve more than 80% average precision on *SO*; on *Math*, *CoPs-QQ* is 1.9% higher than the best competitor wrt average prediction precision. Notice that all the improvements are reported in terms of the absolute precision scores. In general, these results confirm that our *joint* prediction method is effective to predict the voting scores of questions/answers. Specially, our method is better than *CoCQA*. The reason is that *CoCQA* trains two classifiers for both question and answer prediction, but still ignores the correlation between question scores and answer scores. Our method is also better than the *CQA-MR* method, although both *CQA-MR* and our method aim to improve classification performance by *joint* prediction. There might be two major reasons that contribute to such a performance gap between *CoPs* and *CQA-MR*. First, while *CQA-MR* employs voting correlation through the label space (by propagating the labels through user-question–answer graph), our *CoPs* does so through both label space and feature space. Second, our *CoPs* finds a local minimum for Eq. (2); in contrast, *CQA-MR* alternates between propagating labels and maximizing the corresponding conditional likelihood, and therefore it is not clear what overall cost function *CQA-MR* aims to optimize and whether or not the overall procedure converges.



(a) Answer Prediction (b) Question Prediction

Fig. 5. The prediction results (precision) of *CoPs* on the *SO* data. Overall, our methods are better than the compared methods especially for answer prediction.



(a) Answer Prediction (b) Question Prediction

Fig. 6. The prediction results (precision) of *CoPs* on the *Math* data. For both question and answer prediction, most of our methods outperform the compared methods.

(B) The Effectiveness of CoPs at Creation Time. Next, it would be useful if we can predict the voting of the upcoming answers immediately after the questions are posted. For example, we could provide feedback for the questioner if his/her question is unlikely to attract good answers. Our method can enable such prediction by employing the voting correlation over the features that are available at the question's creation time (i.e., *before* any answers actually show up). The result is shown in Fig. 7. The *separate* method cannot work as there are no answers available at the question's creation time. Therefore, we use random guess as the baseline method here. As we can see from the figures, our methods can achieve much higher precision scores (up to 16.2% average precision) than the baseline method for predicting the voting of upcoming answers.

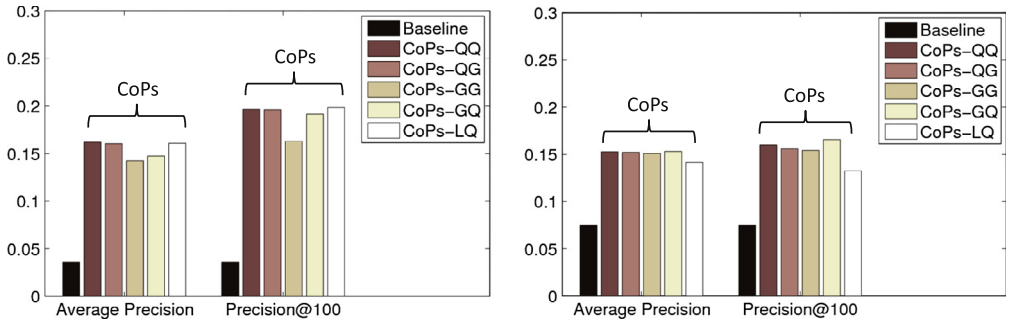
5.3. Efficiency results

Finally, we study the efficiency of *CoPs* by recording the wall-clock time in the training step. We vary the size of the training set, and plot the training time against the total number of questions and answers in the training set. The results of our five variants (i.e., *CoPs-QQ, CoPs-QG,CoPs-GG*, *CoPs-GQ,* and *CoPs-LQ*) are shown in Fig. 8. As we can see, all our five methods scale linearly wrt the size of the training set, which is consistent to the algorithm analysis in Lemma 2. Notice that the *CoPs-QQ* method is much faster than the other four methods because we could derive the closed-form solution for it. Based on these efficiency results, together with the effectiveness results, we recommend *CoPs-QQ* in practice.

6. Related work

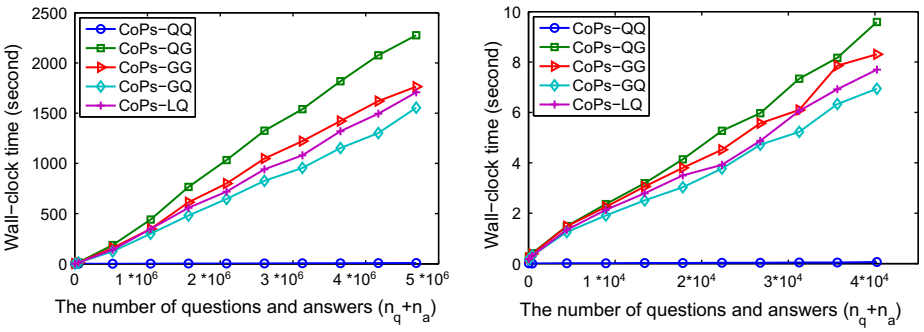
In this section, we briefly review related work.

Measurement of Questions/Answers: There are several types of measurement for questions and answers in CQA. The first one focuses on the *quality* [11,9,27] of question/answer posts. In this branch of work, human annotators are usually needed to manually label the quality of posts. The second type of measurement is the so-called *questioner satisfaction* [16,23,1,29]. This measurement is defined on answers, and it reflects which answer the questioner will probably choose as the accepted answer. The third measurement *question utility* [25] is for questions, and it is defined as the likelihood that a question is repeatedly asked by people. Recently, Anderson et al. propose to predict the *long-lasting value* (i.e., the page-views) [3] of a question and its answers. This measurement views question and its associated answers as an unit.



(a) *SO* data (b) *Math* data

Fig. 7. The prediction results (precision) of *CoPs* for upcoming answers at the creation time of questions. The *Baseline* method stands for the random guess. Our methods are much better than the *Baseline* method.



(a) *SO* data (b) *Math* data

Fig. 8. The wall-clock time of our methods. Our methods scale linearly wrt the total number of questions and answers ($n_q + n_a$).

Although closely related, the *voting score* studied in this paper bears some subtle differences from the above measures. Compared to the quality measure, voting score directly measures how many users find the post beneficial to them. Compared to question utility and questioner satisfaction, voting score can measure both questions and answers. Let us use scientific publication as an analogy to illustrate the difference between voting score and long-lasting value. The number of pageviews of a post can be seen as the number of views/downloads of a research paper, while the voting score can be seen as the number of citations of a paper. Voting score is also studied by several other researchers. For instance, Ravi et al. [21] combine voting score and pageviews to define question quality, Nasehi et al. [18] investigate the high-score code examples in Stack Overflow, and Saha et al. [22] study the causes of the unanswered questions.

Prediction Methods: Here, we review the methods used for predicting a certain type of measurement. Most of existing work treats the prediction of questions or answers as two separate problems. For example, Li et al. [12] explore the interaction of users and topics to predict question quality, which is defined as the combination of user attention, answer attempts and the arrival speed of the best answer. Jeon et al. [11] and Suryanto et al. [27] evaluate the usefulness of answer quality and incorporate it to improve retrieval performance. Our work is different from the above work at the methodology level: instead of treating the prediction as two separate problems, we observe the positive voting correlation between questions and their answers, and propose to jointly predict the voting scores of questions/answers by leveraging such correlation.

Some existing work focuses on the quality prediction of both questions and answers. For example, Agichtein et al. [2] develop a graph-based model to capture the relationships among users; Li et al. [13] adopt the co-training approach to employ both question features and answer features. However, at the methodology level, these two methods still treat question prediction and answer prediction as separate problems. Bian et al. [5] propose to propagate the labels through a user–question–answer graph, so as to tackle the sparsity problem where only a small number of questions/answers are labeled. In contrast, we formulate an optimization problem to penalize the differences between question labels and answer labels.

Empirical Studies: There are many empirical studies on CQA sites. For example, Treude et al. [30] investigate Stack Overflow to identify which types of questions are frequently asked and answered by the programmers. Tausczik and Pennebaker [28] study the correlation between user reputation and post score in MathOverflow. Parnin et al. [20] study whether Stack Overflow can be used as a substitute of API documentation. Barua et al. [4] analyze the text content of the posts in Stack Overflow to discover the current hot topics that software developers are discussing. Mamykina et al. [17] study the successful design choices of Stack Overflow so that the lessons can be reused for other applications. Different from the existing empirical studies, our focus is to quantitatively verify the voting correlation between questions and answers, and we further leverage such correlation to boost the prediction performance.

Other Related Work: There are several pieces of interesting work that are remotely related to our work. Gottipati et al. [8] focus on how to find relevant answers in a software forum when there could be many answers for a single question. Liu et al. [15] propose the problem of CQA site searcher satisfaction, i.e., whether or not the answer in a CQA site satisfies the information searcher using the search engines. Shtok et al. [24] attempt to answer certain new questions by existing answers. Correa and Sureka [6,7] study the closed questions in Stack Overflow. Sung et al. [26] aim to detect the potentially contributive users from recently-joined users. The question routing problem (e.g., how to route the right question to the right answerer) is also an active research area [32,14,10].

7. Conclusions

In this paper, we study the relationship between the voting scores of questions and answers in CQA sites. We start with an empirical study on two CQA datasets where we observe a strong positive voting correlation between questions and their associated answers. Armed with this observation, we next propose a family of algorithms to jointly predict the voting scores

of questions and answers. We conduct extensive experimental evaluations to demonstrate the effectiveness and efficiency. To be specific, our *joint* prediction approaches achieve up to 15.2% net precision improvement over the best competitor in one of the data sets we studied. In addition, the proposed *CoPs* algorithms enable us to predict the voting outcome of an answer *before* it actually appears on the site. Finally, our approaches scale linearly wrt the total number of questions and answers. Future work includes investigating the impact of the question/answer content on such correlation structure as well as its dynamics.

Acknowledgments

This work is supported by the National 973 Program of China (No. 2015CB352202), and the National Natural Science Foundation of China (Nos. 91318301 and 61321491). This material is partially supported by the National Science Foundation under Grant No. IIS1017415, by the Army Research Laboratory under Cooperative Agreement Number W911NF-09-2-0053, by Defense Advanced Research Projects Agency (DARPA) under Contract Numbers W911NF-11-C-0200 and W911NF-12-C-0028, by Region II University Transportation Center under the project number 49997-33 25, by the ARO Young Investigator Program grant with Contract No. W911NF-14-1-0029, by an R&D gift from Northrop Grumman Aerospace Systems, and by the Stony Brook University Office of Vice President for Research. Tao Xie's work is supported in part by a Microsoft Research Award, NSF grants CCF-1349666, CNS-1434582, CCF-1434596, CCF-1434590, CNS-1439481, and NSF of China No. 61228203.

The content of the information in this document does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

Appendix A. Algorithm details

Here, we present the detailed steps in Algorithm 1.

The core of the algorithm is to iteratively update the coefficient vectors (steps 5–6):

$$\begin{aligned}\boldsymbol{\beta}_q &\leftarrow \boldsymbol{\beta}_q - \gamma\frac{\partial\mathcal{L}}{\partial\boldsymbol{\beta}_q}\\ \boldsymbol{\beta}_a &\leftarrow \boldsymbol{\beta}_a - \gamma\frac{\partial\mathcal{L}}{\partial\boldsymbol{\beta}_a}\end{aligned} \tag{.1}$$

where γ is the learning step size, and the partial derivatives can be generally computed as:

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial\boldsymbol{\beta}_q} &= \frac{1}{n_q}\sum_{i=1}^{n_q}\frac{\partial g(\mathbf{X}_{\tilde{q}}(i,:)\boldsymbol{\beta}_q, \mathbf{y}_q(i))}{\partial\boldsymbol{\beta}_q} + \frac{\eta}{n_q}\sum_{i=1}^{n_q}\frac{\partial h(\mathbf{X}_{\tilde{q}}(i,:)\boldsymbol{\beta}_q, \tilde{\mathbf{M}}(i,:)\mathbf{X}_{\tilde{a}}\boldsymbol{\beta}_a)}{\partial\boldsymbol{\beta}_q} + 2\lambda\boldsymbol{\beta}_q\\ \frac{\partial\mathcal{L}}{\partial\boldsymbol{\beta}_a} &= \frac{1}{n_a}\sum_{j=1}^{n_a}\frac{\partial g(\mathbf{X}_{\tilde{a}}(j,:)\boldsymbol{\beta}_a, \mathbf{y}_a(j))}{\partial\boldsymbol{\beta}_a} + \frac{\eta}{n_q}\sum_{i=1}^{n_q}\frac{\partial h(\mathbf{X}_{\tilde{q}}(i,:)\boldsymbol{\beta}_q, \tilde{\mathbf{M}}(i,:)\mathbf{X}_{\tilde{a}}\boldsymbol{\beta}_a)}{\partial\boldsymbol{\beta}_a} + 2\lambda\boldsymbol{\beta}_a\end{aligned} \tag{.2}$$

Notice that we have five variants based on the combination of loss functions, and the partial derivatives for each variants can be computed as follows:

$$\begin{cases}\frac{\partial g_{square}}{\partial\boldsymbol{\beta}_q} = & 2(\mathbf{X}_{\tilde{q}}(i,:)\boldsymbol{\beta}_q - \mathbf{y}_q(i))\mathbf{X}_{\tilde{q}}(i,:)'\\ \frac{\partial g_{square}}{\partial\boldsymbol{\beta}_a} = & 2(\mathbf{X}_{\tilde{a}}(j,:)\boldsymbol{\beta}_a - \mathbf{y}_a(j))\mathbf{X}_{\tilde{a}}(j,:)'\end{cases}$$

$$\begin{cases}\frac{\partial g_{logistic}}{\partial\boldsymbol{\beta}_q} = & \left(\frac{1}{1+\exp(-\mathbf{X}_{\tilde{q}}(i,:)\boldsymbol{\beta}_q)} - \mathbf{y}_q(i)\right)\mathbf{X}_{\tilde{q}}(i,:)'\\ \frac{\partial g_{logistic}}{\partial\boldsymbol{\beta}_a} = & \left(\frac{1}{1+\exp(-\mathbf{X}_{\tilde{a}}(j,:)\boldsymbol{\beta}_a)} - \mathbf{y}_a(j)\right)\mathbf{X}_{\tilde{a}}(j,:)'\end{cases}$$

$$\begin{cases}\frac{\partial g_{sigmoid}}{\partial\boldsymbol{\beta}_q} = & -\frac{\exp(\mathbf{X}_{\tilde{q}}(i,:)\boldsymbol{\beta}_q\mathbf{y}_q(i))}{(1+\exp(\mathbf{X}_{\tilde{q}}(i,:)\boldsymbol{\beta}_q\mathbf{y}_q(i)))^2}\mathbf{y}_q(i)\mathbf{X}_{\tilde{q}}(i,:)'\\ \frac{\partial g_{sigmoid}}{\partial\boldsymbol{\beta}_a} = & -\frac{\exp(\mathbf{X}_{\tilde{a}}(j,:)\boldsymbol{\beta}_a\mathbf{y}_a(j))}{(1+\exp(\mathbf{X}_{\tilde{a}}(j,:)\boldsymbol{\beta}_a\mathbf{y}_a(j)))^2}\mathbf{y}_a(j)\mathbf{X}_{\tilde{a}}(j,:)'\end{cases}$$

$$\begin{cases}\frac{\partial h_{square}}{\partial\boldsymbol{\beta}_q} = & 2(\mathbf{X}_{\tilde{q}}(i,:)\boldsymbol{\beta}_q - \tilde{\mathbf{M}}(i,:)\mathbf{X}_{\tilde{a}}\boldsymbol{\beta}_a)\mathbf{X}_{\tilde{q}}(i,:)'\\ \frac{\partial h_{square}}{\partial\boldsymbol{\beta}_a} = & 2(\mathbf{X}_{\tilde{q}}(i,:)\boldsymbol{\beta}_q - \tilde{\mathbf{M}}(i,:)\mathbf{X}_{\tilde{a}}\boldsymbol{\beta}_a)\mathbf{X}_{\tilde{a}}'\tilde{\mathbf{M}}(i,:)'\end{cases}$$

$$\begin{cases}\frac{\partial h_{sigmoid}}{\partial\boldsymbol{\beta}_q} = & -\frac{\exp(\mathbf{X}_{\tilde{q}}(i,:)\boldsymbol{\beta}_q\tilde{\mathbf{M}}(i,:)\mathbf{X}_{\tilde{a}}\boldsymbol{\beta}_a)}{(1+\exp(\mathbf{X}_{\tilde{q}}(i,:)\boldsymbol{\beta}_q\tilde{\mathbf{M}}(i,:)\mathbf{X}_{\tilde{a}}\boldsymbol{\beta}_a))^2}\tilde{\mathbf{M}}(i,:)\mathbf{X}_{\tilde{a}}\boldsymbol{\beta}_a\mathbf{X}_{\tilde{q}}(i,:)'\\ \frac{\partial h_{sigmoid}}{\partial\boldsymbol{\beta}_a} = & -\frac{\exp(\mathbf{X}_{\tilde{q}}(i,:)\boldsymbol{\beta}_q\tilde{\mathbf{M}}(i,:)\mathbf{X}_{\tilde{a}}\boldsymbol{\beta}_a)}{(1+\exp(\mathbf{X}_{\tilde{q}}(i,:)\boldsymbol{\beta}_q\tilde{\mathbf{M}}(i,:)\mathbf{X}_{\tilde{a}}\boldsymbol{\beta}_a))^2}\mathbf{X}_{\tilde{q}}(i,:)\boldsymbol{\beta}_q\mathbf{X}_{\tilde{a}}'\tilde{\mathbf{M}}(i,:)'\end{cases}$$

(.3)

As mentioned before, we could also adopt the Newton's method to solve Eq. (2). In this method, we only need to substitute γ with the inverse of the second-order partial derivatives. We give the computations of the second-order partial derivatives for each variant as follows:

$$
\begin{cases}
\frac{\partial^2 g_{square}}{\partial \beta_{\tilde{q}}^2} = & 2\mathbf{X}_{\tilde{q}}(i,:)'\mathbf{X}_{\tilde{q}}(i,:) \\
\frac{\partial^2 g_{square}}{\partial \beta_{\tilde{a}}^2} = & 2\mathbf{X}_{\tilde{a}}(j,:)'\mathbf{X}_{\tilde{a}}(j,:)
\end{cases}
$$

$$
\begin{cases}
\frac{\partial^2 g_{logistic}}{\partial \beta_{\tilde{q}}^2} = & \frac{\exp(-\mathbf{X}_{\tilde{q}}(i,:)\boldsymbol{\beta}_q)}{(1+\exp(-\mathbf{X}_{\tilde{q}}(i,:)\boldsymbol{\beta}_q))^2}\mathbf{X}_{\tilde{q}}(i,:)'\mathbf{X}_{\tilde{q}}(i,:) \\
\frac{\partial^2 g_{logistic}}{\partial \beta_{\tilde{a}}^2} = & \frac{\exp(-\mathbf{X}_{\tilde{a}}(j,:)\boldsymbol{\beta}_a)}{(1+\exp(-\mathbf{X}_{\tilde{a}}(j,:)\boldsymbol{\beta}_a))^2}\mathbf{X}_{\tilde{a}}(j,:)'\mathbf{X}_{\tilde{a}}(j,:)
\end{cases}
$$

$$
\begin{cases}
\frac{\partial^2 g_{sigmoid}}{\partial \beta_{\tilde{q}}^2} = & -\frac{\exp(\mathbf{X}_{\tilde{q}}(i,:)\boldsymbol{\beta}_q\mathbf{y}_q(i))(\exp(\mathbf{X}_{\tilde{q}}(i,:)\boldsymbol{\beta}_q\mathbf{y}_q(i))-1)}{(1+\exp(\mathbf{X}_{\tilde{q}}(i,:)\boldsymbol{\beta}_q\mathbf{y}_q(i)))^3} \\
& \mathbf{y}_q(i)^2\mathbf{X}_{\tilde{q}}(i,:)'\mathbf{X}_{\tilde{q}}(i,:) \\
\frac{\partial^2 g_{sigmoid}}{\partial \beta_{\tilde{a}}^2} = & -\frac{\exp(\mathbf{X}_{\tilde{a}}(j,:)\boldsymbol{\beta}_a\mathbf{y}_a(j))(\exp(\mathbf{X}_{\tilde{a}}(j,:)\boldsymbol{\beta}_a\mathbf{y}_a(j))-1)}{(1+\exp(\mathbf{X}_{\tilde{a}}(j,:)\boldsymbol{\beta}_a\mathbf{y}_a(j)))^3} \\
& \mathbf{y}_a(j)^2\mathbf{X}_{\tilde{a}}(j,:)'\mathbf{X}_{\tilde{a}}(j,:)
\end{cases}
$$

$$
\begin{cases}
\frac{\partial^2 h_{square}}{\partial \beta_{\tilde{q}}^2} = & 2\mathbf{X}_{\tilde{q}}(i,:)'\mathbf{X}_{\tilde{q}}(i,:) \\
\frac{\partial^2 h_{square}}{\partial \beta_{\tilde{a}}^2} = & 2\mathbf{X}_{\tilde{a}}'\tilde{\mathbf{M}}(i,:)'\tilde{\mathbf{M}}(i,:)\mathbf{X}_{\tilde{a}}
\end{cases}
$$

$$
\begin{cases}
\frac{\partial^2 h_{sigmoid}}{\partial \beta_{\tilde{q}}^2} = & -\frac{\exp(\mathbf{X}_{\tilde{q}}(i,:)\boldsymbol{\beta}_q\tilde{\mathbf{M}}(i,:)\mathbf{X}_{\tilde{a}}\boldsymbol{\beta}_a)-1}{(1+\exp(\mathbf{X}_{\tilde{q}}(i,:)\boldsymbol{\beta}_q\tilde{\mathbf{M}}(i,:)\mathbf{X}_{\tilde{a}}\boldsymbol{\beta}_a))^3}(\tilde{\mathbf{M}}(i,:)\mathbf{X}_{\tilde{a}}\boldsymbol{\beta}_a)^2 \\
& \exp(\mathbf{X}_{\tilde{q}}(i,:)\boldsymbol{\beta}_q\tilde{\mathbf{M}}(i,:)\mathbf{X}_{\tilde{a}}\boldsymbol{\beta}_a)\mathbf{X}_{\tilde{q}}(i,:)'\mathbf{X}_{\tilde{q}}(i,:) \\
\frac{\partial^2 h_{sigmoid}}{\partial \beta_{\tilde{a}}^2} = & -\frac{\exp(\mathbf{X}_{\tilde{q}}(i,:)\boldsymbol{\beta}_q\tilde{\mathbf{M}}(i,:)\mathbf{X}_{\tilde{a}}\boldsymbol{\beta}_a)-1}{(1+\exp(\mathbf{X}_{\tilde{q}}(i,:)\boldsymbol{\beta}_q\tilde{\mathbf{M}}(i,:)\mathbf{X}_{\tilde{a}}\boldsymbol{\beta}_a))^3}(\mathbf{X}_{\tilde{q}}(i,:)\boldsymbol{\beta}_q)^2 \\
& \exp(\mathbf{X}_{\tilde{q}}(i,:)\boldsymbol{\beta}_q\tilde{\mathbf{M}}(i,:)\mathbf{X}_{\tilde{a}}\boldsymbol{\beta}_a)\mathbf{X}_{\tilde{a}}'\tilde{\mathbf{M}}(i,:)'\tilde{\mathbf{M}}(i,:)\mathbf{X}_{\tilde{a}}
\end{cases}
\tag{.4}
$$

A.1. Proofs for lemmas

Next, we present the proofs for the lemmas.

Proof Sketch for Lemma 1. In each iteration, the algorithm takes steps to the negative of the gradient. Based on the alternating procedure, we have that Algorithm 1 finds a local minimum for the optimization problem in Eq. (2).

For the special case of *CoPs-QQ*, Eq. (4) can be derived by setting the first-order partial derivatives as zeros. □.

Proof of Lemma 2. In Algorithm 1, the time cost for steps 1–2 is $O((n_q+n_a)d)$ because we store $\mathbf{M}$ in its sparse format and there are totally n_a non-zero elements in $\mathbf{M}$. Step 3 needs $O(d)$ time. For steps 5–6, we first compute $\tilde{\mathbf{M}}\mathbf{X}_{\tilde{a}}$ which needs $O(n_a d)$ time. By doing so, we can compute each equation in Eq. (.3) in $O(d)$ time, and thus computing Eq. (.2) with gradient method requires $O((n_q+n_a)d)$ time. If we use Newton's method to update Eq. (.1), we further need to compute the inverse of the second-order partial derivatives in Eq. (.4). This would cost us $O(d^3)$ time. Overall, computing steps 5–6 requires $O((n_q+n_a)d)$ time with gradient method, and $O((n_q+n_a)d+d^3)$ time with Newton's method. Therefore, Algorithm 1 requires $O((n_q+n_a)dm)$ time with gradient descend method and $O((n_q+n_a)dm+d^3m)$ time with Newton's method, which completes the proof. □

Proof of Lemma 3. In Algorithm 1, we need $O((n_q+n_a)d)$ space for input. The space cost for steps 1–2 is also $O((n_q+n_a)d)$. We need $O(d)$ space for step 3. For steps 5–6, we need $O(n_q d)$ space to store $\tilde{\mathbf{M}}\mathbf{X}_{\tilde{a}}$. Then, each equation in Eq. (.3) requires $O(d)$ space. For the Newton's method in Eq. (.4), we need additional $O(d^2)$ space to store the Hessian matrix. Among the different iterations of the algorithm, we can reuse the space from the previous iteration. Therefore, the overall space cost is $O((n_q+n_a)d)$ with gradient descend method and $O((n_q+n_a)d+d^2)$ with Newton's method, which completes the proof. □

References

[1] L. Adamic, J. Zhang, E. Bakshy, M. Ackerman, Knowledge sharing and yahoo answers: everyone knows something, in: WWW, 2008, pp. 665–674.
[2] E. Agichtein, C. Castillo, D. Donato, A. Gionis, G. Mishne, Finding high-quality content in social media, in: WSDM, 2008, pp. 183–194.
[3] A. Anderson, D. Huttenlocher, J. Kleinberg, J. Leskovec, Discovering value from community activity on focused question answering sites: a case study of stack overflow, in: KDD, 2012, pp. 850–858.

[4] A. Barua, S. Thomas, A. Hassan, What are developers talking about? an analysis of topics and trends in stack overflow, Empirical Software Eng. (2012) 1–36
[5] J. Bian, Y. Liu, D. Zhou, E. Agichtein, H. Zha, Learning to recognize reliable users and content in social media with coupled mutual reinforcement, in: WWW, 2009, pp. 51–60.
[6] D. Correa, A. Sureka, Fit or unfit: analysis and prediction of 'closed questions' on stack overflow, in: Proceedings of the first ACM Conference on Online Social Networks, 2013, pp. 201–212.
[7] D. Correa, A. Sureka, Chaff from the wheat: characterization and modeling of deleted questions on stack overflow, in: Proceedings of the 23rd international conference on World wide web. International World Wide Web Conferences Steering Committee, 2014, pp. 631–642.
[8] S. Gottipati, D. Lo, J. Jiang, Finding relevant answers in software forums, in: ASE, 2011, pp. 323–332.
[9] F. Harper, D. Raban, S. Rafaeli, J. Konstan, Predictors of answer quality in online q&a sites, in: CHI, 2008, pp. 865–874.
[10] D. Horowitz, S. Kamvar, The anatomy of a large-scale social search engine, in: WWW, 2010, pp. 431–440.
[11] J. Jeon, W. Croft, J. Lee, S. Park, A framework to predict the quality of answers with non-textual features, in: SIGIR, 2006, pp. 228–235.
[12] B. Li, T. Jin, M.R. Lyu, I. King, B. Mak, Analyzing and predicting question quality in community question answering services, in: WWW Companion, 2012, pp. 775–782.
[13] B. Li, Y. Liu, E. Agichtein, Cocqa: co-training over questions and answers with an application to predicting question subjectivity orientation, in: EMNLP, 2008, pp. 937–946.
[14] W. Li, C. Zhang, S. Hu, G-finder: routing programming questions closer to the experts, in: OOPSLA, 2010, pp. 62–73.
[15] Q. Liu, E. Agichtein, G. Dror, E. Gabrilovich, Y. Maarek, D. Pelleg, I. Szpektor, Predicting web searcher satisfaction with existing community-based answers, in: SIGIR, 2011, pp. 415–424.
[16] Y. Liu, J. Bian, E. Agichtein, Predicting information seeker satisfaction in community question answering, in: SIGIR, 2008, pp. 483–490.
[17] L. Mamykina, B. Manoim, M. Mittal, G. Hripcsak, B. Hartmann, Design lessons from the fastest q&a site in the west, in: CHI, 2011, pp. 2857–2866.
[18] S.M. Nasehi, J. Sillito, F. Maurer, C. Burns, What makes a good code example?: A study of programming q&a in stackoverflow, in: ICSM, 2012, pp. 25–34.
[19] T. Osbourn, Getting the most out of the web, Software IEEE 28 (1) (2011) 96.
[20] C. Parnin, C. Treude, L. Grammel, M. Storey, Crowd documentation: exploring the coverage and the dynamics of api discussions on stack overflow, Georgia Institute of Technology, Tech. Rep., 2012.
[21] S. Ravi, B. Pang, V. Rastogi, R. Kumar, Great question! question quality in community q&a, in: ICWSM, 2014, pp. 426–435.
[22] R.K. Saha, A.K. Saha, D.E. Perry, Toward understanding the causes of unanswered questions in software information sites: a case study of stack overflow, in: FSE, New Ideas Track, 2013.
[23] C. Shah, J. Pomerantz, Evaluating and predicting answer quality in community qa, in: SIGIR, 2010, pp. 411–418.
[24] A. Shtok, G. Dror, Y. Maarek, I. Szpektor, Learning from the past: answering new questions with past answers, in: WWW, 2012, pp. 759–768.
[25] Y. Song, C. Lin, Y. Cao, H. Rim, Question utility: a novel static ranking of question search, in: AAAI, 2008, pp. 1231–1236.
[26] J. Sung, J.-G. Lee, U. Lee, Booming up the long tails: discovering potentially contributive users in community-based question answering services, in: ICWSM, 2013.
[27] M. Suryanto, E. Lim, A. Sun, R. Chiang, Quality-aware collaborative question answering: methods and evaluation, in: WSDM, 2009, pp. 142–151.
[28] Y. Tausczik, J. Pennebaker, Predicting the perceived quality of online mathematics contributions from users' reputations, in: CHI, 2011, pp. 1885–1888.
[29] Q. Tian, P. Zhang, B. Li, Towards predicting the best answers in community-based question-answering services, in: ICWSM, 2013.
[30] C. Treude, O. Barzilay, M. Storey, How do programmers ask and answer questions on the web?, in: ICSE NIER track, 2011, pp. 804–807.
[31] Y. Yao, H. Tong, T. Xie, L. Akoglu, F. Xu, J. Lu, Want a good answer? ask a good question first!, 2013, arXiv preprint arXiv:1311.6876.
[32] Y. Zhou, G. Cong, B. Cui, C. Jensen, J. Yao, Routing questions to the right users in online communities, in: ICDE, 2009, pp. 700–711.