Joint Voting Prediction for Questions and Answers in CQA

Yuan Yao[1], Hanghang Tong[2], Tao Xie[3], Leman Akoglu[4], Feng Xu[1], Jian Lu[1]
[1]State Key Laboratory for Novel Software Technology, Nanjing University, China
[2]City College, CUNY, USA
[3]University of Illinois at Urbana-Champaign, USA
[4]Stony Brook University, USA
yyao@smail.nju.edu.cn, tong@cs.ccny.cuny.edu, taoxie@illinois.edu, leman@cs.stonybrook.edu, {xf, lj}@nju.edu.cn

***Abstract*—Community Question Answering (CQA) sites have become valuable repositories that host a massive volume of human knowledge. How can we detect a high-value answer which clears the doubts of many users? Can we tell the user if the question s/he is posting would attract a good answer? In this paper, we aim to answer these questions from the perspective of the voting outcome by the site users. Our key observation is that the voting score of an answer is strongly positively correlated with that of its question, and such correlation could be in turn used to boost the prediction performance. Armed with this observation, we propose a family of algorithms to *jointly* predict the voting scores of questions and answers soon after they are posted in the CQA sites. Experimental evaluations demonstrate the effectiveness of our approaches.**

I. Introduction

Community Question Answering (CQA) sites have become valuable repositories that host a massive volume of human knowledge. In addition to providing answers to the questioner, CQA sites now serve as knowledge bases for the searching and browsing conducted by a much larger audience. For example, in a software forum called Stack Overflow, programmers can post their programming questions on the forum, and others can propose their answers for these questions. Such questions as well as their associated answers could be valuable and reusable for many other programmers who encounter similar problems. In fact, millions of programmers now use such forums to search for solutions for their programming problems [1].

To maximize the utility of CQA sites, a key task is to characterize and predict the intrinsic value (e.g., quality, impact, etc) of the question/answer posts. This is an essential task for both information producers and consumers. From the perspective of information producers (e.g., who ask or answer questions), it would be helpful to identify the high-value questions in the early stage so that these questions can be recommended to experts for them to answer. From the perspective of information consumers (e.g., who search or browse questions and answers), it would be helpful to highlight high-value questions/answers (e.g., by displaying them more prominently on the site or allowing the search engine to be aware of their value) so that users can easily discover them.

Most of the existing CQA sites allow the site users to vote (e.g., upvote and downvote in Stack Overflow) for a question or an answer. 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 the past, the voting score has been studied in several interesting scenarios (e.g., information quality, user satisfaction, etc; see related work section for details).

In this paper, we aim to study the relationship between the voting scores of questions and those of answers. 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[1] (*SO*), and Mathematics Stack Exchange[2] (*Math*). Our key finding is that the voting score of an answer is indeed strongly positively correlated with that of its question. Such correlation structure consistently exists on both sites. Armed with this observation, we propose a family of 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. Experimental evaluations show that our *joint* prediction approaches achieve up to 15.2% net precision improvement over the best competitor.

The rest of the paper is organized as follows. Section 2 verifies the correlation of voting scores. Sections 3 presents the proposed algorithms. Section 4 presents the experimental results. Section 5 reviews related work, and Section 6 concludes the paper.

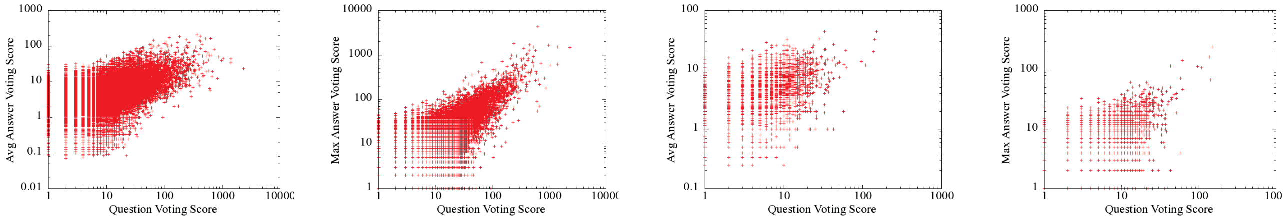
II. Empirical Study

In this section, we perform an empirical study of the voting scores of questions/answers in *SO* and *Math* datasets. They are popular CQA sites for programming and math, respectively. The statistics of the two datasets are summarized in Table I.

We first study the overall correlation between the voting scores of questions and those of their answers. For a given

[1] http://stackoverflow.com/

[2] http://math.stackexchange.com/



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

(b) Maximum answer score vs. Question score in *SO* data(r = 0.6665, p-value < 0.0001)

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

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

Fig. 1. The strong voting correlation between questions and their answers in *SO* and *Math*. r stands for the Pearson correlation coefficient.

TABLE I. 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

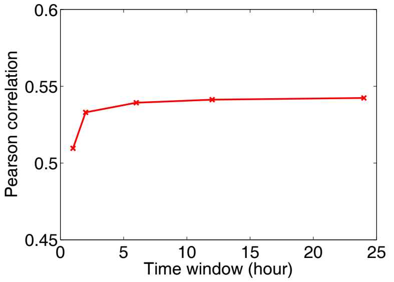


Fig. 2. The voting correlation between questions and their answers over time. The y-axis represents the Pearson correlation coefficient r, and the x-axis represents the time after the question is posted. For all r, p-value < 0.0001.

question, there might be multiple answers. Thus, we report both the highest (i.e., the best answer) and the average voting scores of its answers. The results are shown in Fig. 1, 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 datasets.

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* dataset in Fig. 2. 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.

III. Joint Voting Prediction Approach

In this section, we present our algorithms to jointly predict the *final* voting scores for questions and answers.

Problem Statement. 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 are posted on the site. Our goal is to predict the final voting scores of questions/answers in a short period after they are 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, we should include only the information that is available in the first hour after the question is posted.

For notations, we use $\mathbf{X}_q/\mathbf{X}_a$ to denote the feature matrices for questions/answers where each row contains the feature vector for the corresponding question/answer. The final voting score of questions/answers are denoted by $\mathbf{y}_q/\mathbf{y}_a$. 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 j^{th} answer belongs to the i^{th} question. Similar to Matlab, we also denote the i^{th} row of matrix $\mathbf{M}$ as $\mathbf{M}(i, :)$, and the transpose of a matrix with a prime (i.e., $\mathbf{M}' \equiv \mathbf{M}^T$).

Intuitions and Basic Strategies. We first present the two basic strategies that we explore to leverage the observed voting correlation.

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 $\mathbf{X}_{\tilde{a}} = [\mathbf{X}_a, \mathbf{M}'\mathbf{X}_q]$ 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 $\mathbf{X}_{\tilde{q}} = [\mathbf{X}_q, \tilde{\mathbf{M}}\mathbf{X}_a]$. 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.

The Proposed Approach. Based on the above two strategies (i.e., feature expansion and voting consistency), we propose a new optimization formulation for joint voting prediction of questions/answers:

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

TABLE II. THE FIVE VARIANTS DERIVED FROM EQ. (1).

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

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. (1) 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. (2):

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

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. (1) by setting g and h based on the three loss functions, as shown in Table II. To solve the variants, our key observation is that for each of five cases, the gradient for each term in Eq. (1) exists. This naturally leads to a gradient-descent type of iterative procedure to solve Eq. (1). The detailed algorithms can be found in our tech-report [2].

IV. EXPERIMENTS

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 including the questioners' reputation, the length of the question/answer, the number of comments received. For most of the 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. In this work, we fix this time window as one hour. Detailed feature description can be found in our tech-report [2].

We formulate the task of voting prediction as a binary classification task, where we want to identify the small amount of high-score questions/answers. In particular, we define the posts whose score is no less than 10 as high-score posts in both *SO* and *Math* datasets. This results in 3.4% and 8.7% high-score posts in *SO* and *Math*, respectively. In other words, both datasets 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 [2]. For each dataset, we randomly choose 10% questions and their associated answers as the training set, and use the rest as the test set. 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 $\lambda = 10^{-4}$.

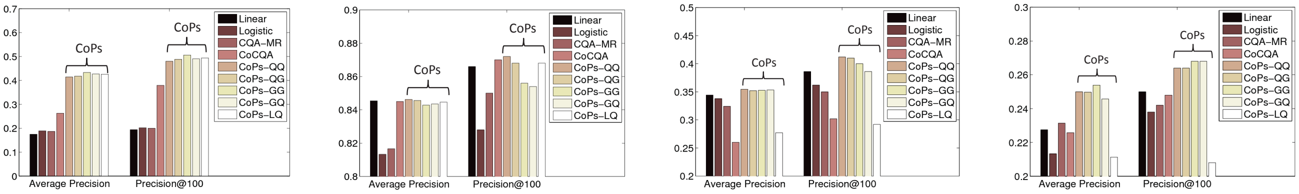
Experimental Results. We compare our methods with several existing methods, including the separate *Linear* regression method, the separate *Logistic* regression method, the *CQA-MR* method [3], and the *CoCQA* method [4]. The results on *SO* and *Math* are shown in Fig. 3. In the figures, we report the precision at 100 as well as the average precision over 10, 50, 100, 150, ..., 400. All the reported results here are the average of 5 experiments.

As we can see, overall, our *CoPs* methods outperform all the compared methods on both datasets. 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. (1); 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.

V. RELATED WORK

Existing measurement for questions and answers in CQA includes *quality* [5], [6], *questioner satisfaction* [7], [8], *question utility* [9], and *long-lasting value* [10]. Although closely related, the *voting score* studied in this paper bears some subtle differences from the above measures. Compared to quality and long-lasting value, 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.

As to the prediction method, most of existing work treats the prediction of questions or answers as two separate problems [5], [6], [11], [4], [12]. As an exception, Bian et al. [3]



(a) Answer Prediction on *SO* data

(b) Question Prediction on *SO* data

(c) Answer Prediction on *Math* data

(d) Question Prediction on *Math* data

Fig. 3. The prediction results (precision) of *CoPs*. Overall, our methods are better than the compared methods.

propose to propagate the labels through 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.

There are several pieces of interesting work that are remotely related to our work. For example, some empirical studies are conducted on CQA sites [13], [14], [15]. Different from these studies, our focus is to quantitatively verify the voting correlation between questions and answers. Other related work includes CQA site searcher satisfaction [16], potentially contributive user detection [17], question-answer matching [18], etc.

VI. 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. Experimental evaluations show that our *joint* prediction approaches achieve up to 15.2% net precision improvement over the best competitor.

VII. Acknowledgments

This work is supported by the National 863 Program of China (No. 2012AA011205), and the National Natural Science Foundation of China (No. 91318301, 61321491, 61100037). This material is partially supported by 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 Number 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.

References

[1] T. Osbourn, "Getting the most out of the web," *Software, IEEE*, vol. 28, no. 1, pp. 96–96, 2011.

[2] Y. Yao, H. Tong, T. Xie, L. Akoglu, F. Xu, and J. Lu, "Want a good answer? ask a good question first!" *arXiv preprint arXiv:1311.6876*, 2013.

[3] J. Bian, Y. Liu, D. Zhou, E. Agichtein, and H. Zha, "Learning to recognize reliable users and content in social media with coupled mutual reinforcement," in *WWW*, 2009, pp. 51–60.

[4] B. Li, Y. Liu, and E. Agichtein, "Cocqa: co-training over questions and answers with an application to predicting question subjectivity orientation," in *EMNLP*, 2008, pp. 937–946.

[5] J. Jeon, W. Croft, J. Lee, and S. Park, "A framework to predict the quality of answers with non-textual features," in *SIGIR*, 2006, pp. 228–235.

[6] M. Suryanto, E. Lim, A. Sun, and R. Chiang, "Quality-aware collaborative question answering: methods and evaluation," in *WSDM*, 2009, pp. 142–151.

[7] Y. Liu, J. Bian, and E. Agichtein, "Predicting information seeker satisfaction in community question answering," in *SIGIR*, 2008, pp. 483–490.

[8] Q. Tian, P. Zhang, and B. Li, "Towards predicting the best answers in community-based question-answering services," in *ICWSM*, 2013.

[9] Y. Song, C. Lin, Y. Cao, and H. Rim, "Question utility: A novel static ranking of question search," in *AAAI*, 2008, pp. 1231–1236.

[10] A. Anderson, D. Huttenlocher, J. Kleinberg, and J. Leskovec, "Discovering value from community activity on focused question answering sites: a case study of stack overflow," in *KDD*, 2012, pp. 850–858.

[11] E. Agichtein, C. Castillo, D. Donato, A. Gionis, and G. Mishne, "Finding high-quality content in social media," in *WSDM*, 2008, pp. 183–194.

[12] B. Li, T. Jin, M. R. Lyu, I. King, and B. Mak, "Analyzing and predicting question quality in community question answering services," in *WWW Companion*, 2012, pp. 775–782.

[13] C. Treude, O. Barzilay, and M. Storey, "How do programmers ask and answer questions on the web?" in *ICSE NIER track*, 2011, pp. 804–807.

[14] A. Barua, S. Thomas, and A. Hassan, "What are developers talking about? an analysis of topics and trends in stack overflow," *Empirical Software Engineering*, pp. 1–36, 2012.

[15] L. Mamykina, B. Manoim, M. Mittal, G. Hripcsak, and B. Hartmann, "Design lessons from the fastest q&a site in the west," in *CHI*, 2011, pp. 2857–2866.

[16] Q. Liu, E. Agichtein, G. Dror, E. Gabrilovich, Y. Maarek, D. Pelleg, and I. Szpektor, "Predicting web searcher satisfaction with existing community-based answers," in *SIGIR*, 2011, pp. 415–424.

[17] J. Sung, J.-G. Lee, and U. Lee, "Booming up the long tails: Discovering potentially contributive users in community-based question answering services," in *ICWSM*, 2013.

[18] A. Shtok, G. Dror, Y. Maarek, and I. Szpektor, "Learning from the past: answering new questions with past answers," in *WWW*, 2012, pp. 759–768.