Weakly Hierarchical Lasso based Learning to Rank in Best Answer Prediction

Conference Paper · August 2016

DOI: 10.1109/ASONAM.2016.7752250

2 authors, including:

Qiongjie Tian
Arizona State University
12 PUBLICATIONS **172** CITATIONS

SEE PROFILE

Weakly Hierarchical Lasso based Learning to Rank in Best Answer Prediction

Qiongjie Tian
Computer Science and Engineering
Arizona State University
Email: qiongjie.tian@asu.edu

Baoxin Li
Computer Science and Engineering
Arizona State University
Email: baoxin.li@asu.edu

***Abstract*—In community question and answering sites, pairs of questions and their high-quality answers (like best answers selected by askers) can be valuable knowledge available to others. However lots of questions receive multiple answers but askers do not label either one as the accepted or best one even when some replies answer their questions. To solve this problem, high-quality answer prediction or best answer prediction has been one of important topics in social media. These user-generated answers often consist of multiple "views", each capturing different (albeit related) information (e.g., expertise of the asker, length of the answer, etc.). Such views interact with each other in complex manners that should carry a lot of information for distinguishing a potential best answer from others. Little existing work has exploited such interactions for better prediction. To explicitly model these information, we propose a new learning-to-rank method, ranking support vector machine (RankSVM) with weakly hierarchical lasso in this paper. The evaluation of the approach was done using data from Stack Overflow. Experimental results demonstrate that the proposed approach has superior performance compared with approaches in state-of-the-art.**

I. Introduction

In the era of Internet and social media, community question and answering (CQA) sites, like Baidu Zhidao[1], Yahoo! Answers[2] and StackOverflow[3], are seeing phenomenal growth. As one form of user-generate content, data from CQA sites are typically very noisy, which does not lead to ready usage either by humans or by computers. Consequently, how to extract useful information from the noisy CQA data to form valuable knowledge base has become an important research task [1]. One popular task on this regard is best answer prediction, on which our paper focuses.

Given a question with multiple answers, one way to solve best answer prediction is to reformulate it into a binary classification problem which is whether, in a question-answer pair, the answer is the best one or not. There have been some research efforts in this setting like [2], [3]. In these efforts, features were extracted from different views of the data to generate a good representation for the question-answer pairs, and the final feature vector was formed by concatenating them together. As a result, each feature dimension carries some information of the CQA data. But there are a couple of limitations inherent to these existing techniques. First, a binary classifier is not natural to this research problem, which often involves multiple answers for one given question. It is possible for a trained classifier to declare many or *even all* answers are the best ones (if they happen to lead to feature vectors lying on the positive side of the decision boundary). Also it is counter-intuitive as a human user would normally compare all received answers and decide on a single best one. The binary classification does not model directly on the difference of multiple answers, compared with learning-to-rank techniques. Second, the interaction between features from different views may carry a lot of information for distinguishing a potential best answer from others, however current existing methods do not readily support incorporation of such interactions, which by itself is a challenging task.

In anther setting, best answer prediction is modeled as one ranking problem, which is conceptually more intuitive. This kind of modeling results from the fact that the best answer to one question is defined/discovered relatively by comparing it with all the other given answers. A ranking-based setting may benefit even more from considering the latent interactions between features designed from different views of the CQA data. Unfortunately, similar to the binary-classification cases, the existing learning-to-rank techniques have not attempted to explicitly to model such interactions among different views of the data [4][5][6].

In this paper, we focus on how to incorporate the interaction structure of features into one existing algorithm framework to improve the performance of best answer prediction. Similar to [5][7], we adopt the learning-to-rank formulation for its natural match to the prediction problem. Considering the interaction structure (or the hierarchical structure of feature dimensions in our study) and the ranking framework, we propose a new learning-to-rank formulation based on weakly hierarchical lasso.

The contributions of our work are summarized as follows: Firstly, we propose a new RankSVM model by constructing the weakly hierarchical structure between features from different views. Secondly, to solve the new formulation, we propose an efficient algorithm and evaluate via experiments its efficiency and effectiveness with comparisons with other existing methods.

[1]http://zhidao.baidu.com/
[2]https://answers.yahoo.com/
[3]http://stackoverflow.com/

IEEE/ACM ASONAM 2016, August 18-21, 2016, San Francisco, CA, USA 978-1-5090-2846-7/16/$31.00

II. Related Work

In this section, we review briefly related research on community question and answering, and discuss the difference between the reviewed work and our proposed method.

A. Content Quality Analysis

Compared with traditional on-line search, as one supplementary approach to solving our daily problems, CQA sites contain a lot of valuable knowledge. Thus, since the first CQA site was launched, finding high quality content from these sites has become important. For example some early work was done in [8] where Jiwoon Jeon *et al.* crawled data from Naver Q&A site and manually labeled each pair of questions and their corresponding answers as *bad, medium, good*. They proposed to use non-textual features to represent each question-answer pair and used kernel density estimation and the maximum entropy approach to model the problem of answer quality. To have a better representation of questions and answers on CQA sites, more sources of information were used to extract new features like interactions between questions and answers and users, as studied in [2], where Eugene Agichtein *et al.* proposed to use non-content information to model question and answer pairs on CQA sites including the interaction features. Then different classifiers like support vector machine, log-linear classifier and stochastic gradient boosted trees were applied to learn the prediction model, whose efficiency and effectiveness were evaluated using data from Yahoo! Answers. The importance of social information for predicting answer quality was studied in [3], where Chirag Shah *et al.* found the importance of user information by studying the quality labeled manually. Besides research on the answer quality, question quality is also studied. In [9], Baichuan Li *et al.* worked on the question quality prediction problem. They first studied what factors may affect question quality and then proposed a model termed Mutual Reinforcement-based Label Propagation to predict question quality. In [10], it was found that the voting scores of questions have a strong positive correlation with that of the corresponding answers and they proposed a set of co-prediction algorithms to predict the voting scores of questions and answers.

The above work focused on content quality prediction (question quality and answer quality), which is modeled as one classification problem. These existing efforts mainly focused on finding a better representation of the data by introducing various features to facilitate the prediction problem.

B. Best Answer Prediction and Answer Ranking

Pairs of questions and their best answers can be easily used to answer similar questions, as the research in [11] shows. With the fast growth of CQA sites, there are a lot of questions which have high quality answers but no best ones eventually marked. To this end, a lot of research efforts have been devoted to best answer prediction and answer ranking. In [12], Lada Adamic *et al.* analyzed Yahoo! Answers for best answer prediction. They used simple four-dimensional features and reported that the length of answers is the most important factor of answer quality. The problem they are worked on is to predict whether a given answer is the best one of the given question. They did not consider interaction information like relationship between questions and answers and users. It is not natural to model best answer prediction as a classification problem since the best answer is relatively defined. Thus there have been a lot of efforts on modeling best answer prediction as a ranking problem. In [13], Mihar Surdeanu *et al.* proposed a ranking model for non-factoid questions and studied whether ranking algorithms can be used to rank answers for given questions. They also showed the importance of different features in the answer ranking problem. This work was further extended in [14]. Instead of simply applying learning to rank algorithms, some researchers worked on improving the performance by using piggybacking and ranking aggregation techniques. In [7], Felix Hieber *et al.* applied RankSVM algorithms to best answer prediction with piggybacking being used to improve the performance. In their work, interaction features were used, like the similarity between questions and answers. Piggybacking is used to for obtaining a better representation of the questions so that similarity between the questions and answers can help improve the ranking performance of RankSVM. One example work to use ranking aggregation is [15], where Arvind Agarwal *et al.* made a comparison between different learning to rank algorithms and proposed to use ranking aggregation techniques to improve them. But that work focused on the factoid question and answers instead of CQA. In contrast, our work employs hierarchical interactions in the feature space.

There are also some efforts on studying the influence of different combinations of features on the prediction accuracy and also comparison across different CQA sites [16]. Point-wise ranking techniques were also used to rank answers to each question. In [4], Daniel Dalip *et al.* assumed that the voting scores to be the quality scores of answers. Then random forest was used to model the relationship between the scores and features. The final predicted rating scores were used to rank each questions. To evaluate the performance, normalized discounted cumulative gain at top k (NDCG@K) is used. However, there is noise in the rating scores as shown in [17], and thus in our work we do not use this assumption. The information between answers to each question may help capture the *relative* information for better prediction, as shown in [18], where Tian *et al.* proposed to extract features from the context information between answers to each question. There are many other efforts on finding/defining new features for best answer prediction. For example, temporal features are proposed in [5].

One common observation in the most of the existing work is that, when new features are derived, all of them are concatenated to one vector to be the final feature vector. For example, in [12], these features are used: reply length, thread length, the total number of best answers of one user, the total number of replies one user has. They can be denoted as x_1, x_2, x_3, x_4. Then the final feature vectors are the simple concatenation of these features which are (x_1, x_2, x_3, x_4). In

our work, we focus on proposing a new model which can capture the feature interactions based on hierarchical lasso.

III. Problem Description and Formulation

The research problem in this paper is formally defined as follows: given a question with all of its received answers, to predict which one is the best one. To select the best answer, one has to compare it with the others, so that the best answer is relatively defined. Thus instead of using the classification framework, we employ the learning-to-rank strategy. The basis of our proposed approach is RankSVM [6]. While existing work focuses on designing new features, we study this prediction problem from the following angle: modeling the interaction of features from different views of data beyond simple concatenation of them. To achieve this goal, we employ weakly hierarchical lasso [19] in constructing a new ranking model.

Notations of this paper are described in the following. Denote a dataset with N questions as $\{q_i,\ i \in \{1, \cdots, N\}\}$. For each question q_i, it receives a group of answers which are $\{A_{i,j},\ j \in \{1, \cdots, M_i\}\}$ where M_i is the total number of answers to q_i. The feature vector $x_{i,j} \in \mathbb{R}^{1\times d}$ is used to represent the j^{th} answer to the i^{th} question. Moreover, the k^{th} dimension of one feature vector $x_{i,j}$ is defined as $x_{i,j,k}$ where $k \in \{1, \cdots, d\}$. $x_{i,j}$ is the simple concatenation of features extracted from different views of our problem, as done in the existing work. It is named as the *main effect*. Then for each $x_{i,j}$, we compute the second-order interaction which is denoted as $z_{i,j} \in \mathbb{R}^{1\times d^2}$, which is called the second-order *interaction term*. The final feature vector by considering the main effect and the interaction term is denoted as $\hat{x}(i,j) = [x_{i,j}, z_{i,j}] \in \mathbb{R}^{1\times(d+d^2)}$. The interaction term is defined as follows (see Eqn.1):

$$\begin{aligned} z_{i,j} &= [z_{i,j}^{(1)}, z_{i,j}^{(2)}, \cdots, z_{i,j}^{(d)}] \\ z_{i,j}^{(m)} &= [x_{i,j,m}\cdot x_{i,j,1},\ x_{i,j,m}\cdot x_{i,j,2},\ \cdots, x_{i,j,m}\cdot x_{i,j,d}] \end{aligned} \tag{1}$$

where $i \in \{1, \cdots, N\}$, $j \in \{1, \cdots, M_i\}$ and $m \in \{1, \cdots, d\}$.

In our work, instead of classification methods, learning-to-rank techniques are used to model the *relativeness* of the best answers. Each relatively ranked pair is represented as $(q_i,\ A_{i,j_1},\ A_{i,j_2})$ where the quality of A_{i,j_1} is higher than that of A_{i,j_2}. For simplicity, we may use (i, j_1, j_2) as the short version of $(q_i,\ A_{i,j_1},\ A_{i,j_2})$ in the following equations. The set P_i contains all these pairs of answers to the question q_i. Furthermore, the entire set of these relatively ranked pairs is denoted as P in Eqn.2.

$$P = \bigcup_{i\in\{1,\cdots,N\}} P_i \tag{2}$$

RankSVM, as one state-of-the-art pair-wise learning-to-rank algorithm used in best answer prediction [5][7], is used as the basic building block of our new ranking model.

The RankSVM formulation is given below (Eqn. 3):

$$\begin{aligned} \min_{w\in\mathbb{R}^{d\times 1}} \quad & \frac{1}{2}\|w\|_2^2 + C\sum \xi_{i,j_1,j_2} \\ \text{s.t.} \quad & S_1(i,j_1) \geq S_1(i,j_2) + 1 - \xi_{i,j_1,j_2}, \quad \forall(i,j_1,j_2) \\ & \xi_{i,j_1,j_2} \geq 0, \quad \forall(i,j_1,j_2) \end{aligned} \tag{3}$$

where (i, j_1, j_2) is one ranked QA pair in P and $S(i,j)$ is the quality score function of the j^{th} answer to q_i and defined in Eqn.4.

$$S_1(i,j) = x_{i,j}\ w + w_0 \tag{4}$$

where $w_0 \in \mathbb{R}$.

To improve the performance of RankSVM, our model involves the second-order interactions via constructing one weakly hierarchical structure in the feature space. The formulation of the new ranking model is shown in Eqn.5. Compared with the existing work, we model the latent interaction structure between features from different views of the data, instead of simple concatenation. The hierarchical structure of the feature space is constructed through the first group of constraints (a.k.a $\|Q_{.,j}\|_1 \leq |w_j|, j \in \{1, \cdots, d\}$) in Eqn.5.

$$\begin{aligned} \min_{\substack{w\in\mathbb{R}^{d\times 1},\\ Q\in\mathbb{R}^{d\times d}}} \quad & \|w\|_1 + \frac{1}{2}\|Q\|_1 + C\sum_{(i,j_1,j_2)\in P} \xi_{i,j_1,j_2} \\ \text{s.t.} \quad & \|Q_{.,j}\|_1 \leq |w_j|, \quad j \in \{1, \cdots, d\} \\ & \xi_{i,j_1,j_2} \geq 0, \quad \forall(i,j_1,j_2) \in P \\ & S(i,j_1) > S(i,j_2) + 1 - \xi_{i,j_1,j_2}, \quad \forall(i,j_1,j_2) \in P \end{aligned} \tag{5}$$

where $Q_{.,j}$ is the j^{th} column of Q, $\|Q\|_1 = \sum_i \sum_j |Q_{i,j}|$ and $S(\cdot,\cdot)$ is the ranking score for each answer to one question defined in Eqn.6. For example $S(i,j)$ is the ranking score for answer $A_{i,j}$ to q_i.

$$S(i,j) = x_{i,j}w + \frac{1}{2}z_{i,j}\ vec(Q) + w_0 \tag{6}$$

where $vec(Q)$ is the vectorized version of Q and $z_{i,j}$ is shown in Eqn.1 and $w_0 \in \mathbb{R}$.

To help illustrating the proposed model, we depict the hierarchical structure based on one example shown in Figure 1, in which we only show three features: the length of the answer (A_{len}), the number of URLs in the answer (N_{url}), the number of pictures used in the answer (N_{pic}). In this illustration, we can see that the upper layer contains all main effects (a.k.a $x_{i,j}$) while the second layer shows the interaction terms (a.k.a $z_{i,j}$ in Eqn.1) excluding the square values of themselves. When one term contributes to the objective function, no matter it belongs to main effects or interaction terms, its corresponding coefficient is set to be non-zero. For each interaction term, if it contributes to the objective function, then at least one of its corresponding main effects contributes to the objective function. Satisfying these hierarchical constraints, it is easy for us to conclude that the interaction terms contribute less than their corresponding main effects. Specifically, in this figure, if the coefficient of $A_{len}\cdot N_{url}$ is non-zero, then the coefficient of A_{len} is non-zero but that of N_{url} can be zero.

From Eqn. 5, the weakly hierarchical lasso is involved via the first group of constraints (a.k.a $\|Q_{.,j}\|_1 \leq |w_j|, j \in \{1, \cdots, d\}$).

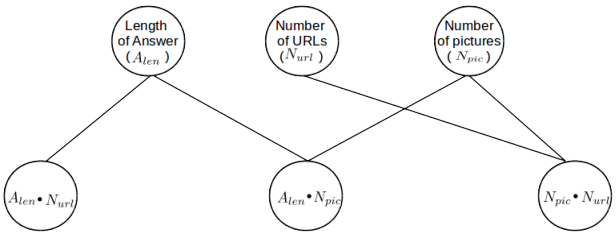

Fig. 1: One illustration to show hierarchical structure in the feature space, where "$\cdot$" represents the scalar multiplication. The first layer contains the main effect, while the second layer consists of the 2^{nd} order of interaction.

IV. Solving the Proposed Model

To develop a solution to our proposed model in Eqn. 5, we first reformulate the problem as follows. Consider this group of constraints (Eqn.7) in the proposed model in Eqn. 5.

$$S_{i,j_1} > S_{i,j_2} + 1 - \xi_{i,j_1,j_2} \tag{7}$$

Together with Eqn.6, we have the following computation:

$$\begin{aligned} S_{i,j_1} &> S_{i,j_2} + 1 - \xi_{i,j_1,j_2} \\ S_{i,j_1} &= x_{i,j_1} w + \frac{1}{2} z_{i,j_1}\ vec(Q) + w_0 \\ S_{i,j_2} &= x_{i,j_2} w + \frac{1}{2} z_{i,j_2}\ vec(Q) + w_0 \end{aligned} \tag{8}$$

If we assume the relatively ranked pair $(q_i, A_{i,j_1}, A_{i,j_2})$ is the m^{th} element in the set P of Eqn.2, then Eqn.8 can be simplified and the following is obtained:

$$\tilde{x}_m w + \frac{1}{2} \tilde{z}_m \cdot vec(Q) > 1 - \tilde{\xi}_m \tag{9}$$

where $\tilde{x}_m$, $\tilde{z}_m$ should satisfy the following constraints in Eqn.10.

$$\begin{aligned} \tilde{x}_m &= x_{i,j_1} - x_{i,j_2} \\ \tilde{z}_m &= z_{i,j_1} - z_{i,j_2} \end{aligned} \tag{10}$$

As a result, Eqn.5 is converted to the following:

$$\begin{aligned} \min_{w,Q} \quad & \|w\|_1 + \frac{1}{2}\|Q\|_1 + C \sum_{m \in \{1,\cdots,|P|\}} \tilde{\xi}_m \\ \text{s.t.} \quad & \tilde{x}_m w + \frac{1}{2} \tilde{z}_m \cdot vec(Q) > 1 - \tilde{\xi}_m, \quad m \in \{1, \cdots, |P|\} \\ & \|Q_{.,j}\|_1 \leq |w_j|, \quad j \in \{1, \cdots, d\} \\ & \tilde{\xi}_m \geq 0, \quad m \in \{1, \cdots, |P|\} \end{aligned} \tag{11}$$

where $|P|$ is the size of the set P.

Now we can reformulate Eqn.11 into Eqn.12:

$$\begin{aligned} \min_{w,Q} \quad & \|w\|_1 + \frac{1}{2}\|Q\|_1 + C \cdot L(w, Q) \\ \text{s.t.} \quad & \|Q_{.,j}\|_1 \leq |w_j|, \quad j \in \{1, \cdots, d\} \end{aligned} \tag{12}$$

where $L(w, Q)$ is given in the following:

$$L(w, Q) = \sum_{m=1}^{|P|} \max(0, 1 - (\tilde{x}_m w + \frac{1}{2} \tilde{z}_m\ vec(Q)))^2 \tag{13}$$

Set $\lambda = \frac{1}{C}$, the final model is obtain as given in Eqn.14

$$\begin{aligned} \min_{w,Q} \quad & L(w, Q) + \lambda \cdot \|w\|_1 + \frac{\lambda}{2}\|Q\|_1 \\ \text{s.t.} \quad & \|Q_{.,j}\|_1 \leq |w_j|, \quad j \in \{1, \cdots, d\} \end{aligned} \tag{14}$$

To this point, our objective function has been reformulated into the standard form as in the weakly hierarchical lasso problem defined in [19] and [20].

To solve Eqn. 14, the scheme in [20] can be applied since it can directly solve the weakly hierarchical lasso without adding more penalty compared with approach in [19]. Since the optimization process in [20] is based on a general iterative shrinkage and thresholding algorithm (GIST) in [21], before we use the method in [20], we need to prove that $L(w, Q)$ in Eqn. 14 is continuously differentiable with Lipschitz continuous gradient.

Before proceeding with the proof, we introduce following notations:

$$\begin{aligned} \hat{x} &= (\tilde{x}, \tilde{z}) \\ \hat{w} &= \begin{pmatrix} w \\ \frac{1}{2} vec(Q) \end{pmatrix} \end{aligned} \tag{15}$$

As a consequence, $\hat{x} \in \mathbb{R}^{1 \times (d + d \cdot d)}$ and $\hat{w} \in \mathbb{R}^{(d + d \cdot d) \times 1}$. $L(w, Q)$ is converted from Eqn.13 as Eqn.16.

$$\hat{L}(\hat{w}) = \sum_{m \in \{1,\cdots,|P|\}} \max(0, 1 - \hat{x}_m \cdot \hat{w})^2 \tag{16}$$

To show $\hat{L}(\hat{w})$ is differentiable with Lipschitz continuous gradient, this requirement needs to be satisfied: there exists a positive constant β such that

$$\|\frac{d\hat{L}}{d\hat{w}}(w_1) - \frac{d\hat{L}}{d\hat{w}}(w_2)\|_2 \leq \beta \|w_1 - w_2\|_2 \tag{17}$$

Let us first consider one additive component of $\hat{L}(\hat{w})$. The point-wise maximum function can be written as Eqn.18.

$$\begin{aligned} l(\hat{w}) &= \max(0, 1 - \hat{x}_m \cdot \hat{w})^2 \\ &= \begin{cases} 0 & \text{if } 1 - \hat{x}_m \cdot \hat{w} < 0 \\ (1 - \hat{x}_m \cdot \hat{w})^2 & \text{if } 1 - \hat{x}_m \cdot \hat{w} \geq 0 \end{cases} \end{aligned} \tag{18}$$

It is easy to see that when $w_1, w_2 \in \{w | 1 - \hat{x}_m \cdot w < 0\}$ and $w_1, w_2 \in \{w | 1 - \hat{x}_m \cdot w \geq 0\}$, Eqn.17 is satisfied. Now considering $w_1 \in \{w | 1 - \hat{x}_m \cdot w < 0\}, w_2 \in \{w | 1 - \hat{x}_m \cdot w > 0\}$, it is easy to see that the left part of Eqn.17 becomes $\|(1 - \hat{x} \cdot w_2)\hat{x}_m\|$. Moreover, define $\hat{w}^*$ as $1 - \hat{x}_m \cdot w^* = 0$ and

this inequality is satisfied: $\|w_1 - w_2\| \geq \|w^* - w_2\|$. Now to obtain the constant β, the following induction is performed:

$$\begin{aligned}
&\|(1 - \hat{x}_m \cdot w_2)\hat{x}_m\| \leq \beta \|w_1 - w_2\| \\
\Leftarrow &\frac{\|(1 - \hat{x}_m \cdot w_2)\hat{x}_m\|}{\|w_1 - w_2\|} \leq \beta \\
\Leftarrow &\frac{\|(1 - \hat{x}_m \cdot w_2)\|\|\hat{x}_m\|}{\|w^* - w_2\|} \leq \beta \\
\Leftarrow &\frac{\|(1 - \hat{x}_m \cdot w_2)\|\|\hat{x}_m\|^2}{\|w^* - w_2\|\|\hat{x}_m\|} \leq \beta \\
\Leftarrow &\frac{\|(1 - \hat{x}_m \cdot w_2)\|\|\hat{x}_m\|^2}{\|1 - \hat{x}_m \cdot w_2\|} \leq \beta \\
\Leftarrow &\beta \geq \|\hat{x}_m\|^2
\end{aligned} \tag{19}$$

Similarly, it is easy to obtain that $\beta \geq \|\hat{x}_m\|^2$ also satisfies the case where $w_2 \in \{w|1 - \hat{x}_m \cdot w < 0\}, w_1 \in \{w|1 - \hat{x}_m \cdot w > 0\}$. Thus, there exists a proper positive constant β so that $l(\hat{w})$ meets the requirement Eqn. 17. In conclusion, $l(\hat{w})$ is continuously differentiable with Lipschitz continuous gradient. With this result, we will further introduce and prove the following lemma, together with which we will able to show the desired property for $L(w, Q)$ is satisfied.

Lemma IV.1. *For each function $f(w)_i, i \in \{1, \cdots, N\}$ which is continuously differentiable with Lipschitz continuous gradient, their summation $f(w) = \sum_{i=1}^{N} f_i(w)$ is continuously differentiable with Lipschitz continuous gradient.*

Proof.

$$\begin{aligned}
&\|\frac{d}{dw}f(w_1) - \frac{d}{dw}f(w_2)\| \\
= &\|\sum_{i=1}^{N}\frac{d}{dw}f_i(w_1) - \sum_{i=1}^{N}\frac{d}{dw}f_i(w_2)\| \\
= &\|\sum_{i=1}^{N}(\frac{d}{dw}f_i(w_1) - \frac{d}{dw}f_i(w_2))\| \\
\leq &\sum_{i=1}^{N}\|\frac{d}{dw}f_i(w_1) - \frac{d}{dw}f_i(w_2)\| \\
\leq &\beta\|w_1 - w_2\|
\end{aligned} \tag{20}$$

Denote that there exists positive constant β_i such that $f_i(w)$ satisfies Eqn.17 where $i \in \{1, \cdots, N\}$. Thus Eqn.20 is valid when β meets this requirement:

$$\beta = \max_i \beta_i \tag{21}$$

□

Since $\max(0, 1 - \hat{x}_m \cdot \hat{w})^2$ satisfies Eqn. 17 and $\hat{L}(\hat{w}) = \sum_{m \in \{1, \cdots, |P|\}} \max(0, 1 - \hat{x}_m \cdot \hat{w})^2$, according to Lemma IV.1, $\hat{L}(\hat{w})$ satisfies Eqn. 17, same as $L(w, Q)$ defined in Eqn. 13. Thus, $L(w, Q)$ is continuously differentiable with Lipschitz continuous gradient.

Now it is feasible to apply the algorithm in [20] to solve Eqn.14 which is equivalent to solving this proximal operator problem of Eqn.22.

$$\begin{aligned}
(w^{(k+1)}, Q^{(k+1)}) = \arg\min_{w,Q} &\frac{1}{2}\|w - v^{(k)}\|_2^2 + \frac{1}{2}\|Q - U^{(k)}\|_2^2 \\
&+ \frac{1}{t^{(k)}}(\lambda\|w\|_1 + \frac{\lambda}{2}\|Q\|_1) \\
\text{s.t.} \quad &\|Q_{.,j}\|_1 \leq |w_j| \quad \forall j \in \{1, \cdots, d\}
\end{aligned} \tag{22}$$

where $v^{(k)}, U^{(k)}$ are defined as follows:

$$v^{(k)} = w^{(k)} - \frac{1}{t^{(k)}} \cdot \bigtriangledown_w L(w^{(k)}, Q^{(k)}) \tag{23}$$

$$U^{(k)} = U^{(k)} - \frac{1}{t^{(k)}} \cdot \bigtriangledown_Q L(w^{(k)}, Q^{(k)}) \tag{24}$$

where $t^{(k)} > 0$ which is the step size.

Considering w, Q are products of their signs and also absolute values, Eqn.22 can be re-written into Eqn.25.

$$\begin{aligned}
(w^{(k+1)}, Q^{(k+1)}) = \arg\min_{w,Q} &\frac{1}{2}\|w - v^{(k)}\|_2^2 + \frac{1}{2}\|Q - U^{(k)}\|_2^2 \\
&+ \frac{1}{t^{(k)}}(\lambda\|w\|_1 + \frac{\lambda}{2}\|Q\|_1) \\
\text{s.t.} \quad &\tilde{Q_{.,j}} \leq \tilde{w_j} \quad \forall j
\end{aligned} \tag{25}$$

where $Q_{.,j} = sign(Q_{.,j})\ \tilde{Q_{.,j}}$ and $w_j = sign(w_j)\ \tilde{w_j}$. The above equation can be solved in a closed form as proved in [20]. The pseudocode of our entire algorithm is shown in the following. which is summarized in Algorithm 1.

Algorithm 1 The pseudo-code to solve our model

```
INPUT: data matrix X and ranking information of all data
OUTPUT: model parameters w and Q
BEGIN:
    compute the set P based on Eqn.2.
    compute the data difference {x̃_m, m ∈ {1, ··· , |P|}}
and {z̃_m, m ∈ {1, ··· , |P|}} as Eqn.10.
    provide initial values for w and Q.
    choose one t via BB Rule [22].
while w, Q satisfy the stop criteria do
    while t^k does not satisfy the stop criteria do
        update v^k according to Eqn.23.
        update U^k according to Eqn.24.
        obtain new w^(k+1) and Q^(k+1) based on Eqn.25,
which can be in the closed form as [20].
        update the step size t^(k) = α * t^(k) where α is the
constant update ratio.
    end while
    k = k + 1;
end while
```

V. Experiments

In this section, we present experimental results on Stack Overflow to show the performance of our proposed model and the comparison with existing methods.

A. Data Description

Founded in 2008, StackOverflow is active and well maintained. On this site, users can post questions and everyone can provide answers even including the askers. For each question and each answer, users can comment on it. For one question or answer, users can vote up or down based on its quality except the user who posts it. For one comment, users can only vote up if they think the comment is useful, but cannot vote down. Same as one question or one answer, the one cannot vote up his or her own comments. For one question or one answer, it can receive up-votes and also down-votes. Then the number of up-votes minus the number of down-votes is the vote score. It is easy to see that the vote score are integers and can be negative.

Each question can receive multiple answers and only the asker can decide which one can be marked as the *accepted answer* which we call the *best answer*. This choice is not permanent, which means the asker can change his or her mind at any time and mark another answer as the *best answer*. There is one fact we need to point out. One question may receive multiple correct answers but only one of them can be marked as the *best answer*. So the *best answer* has the relatively best quality instead of absolutely best one. This is the reason why we use the learning to rank techniques instead of the classification methods. For users, they can earn reputations if their posts (e.g. questions ,answers, and comments) obtain upvotes or answers are accepted or suggestions on editing others' posts are accepted. Otherwise, they lose reputations if their posts receive downvotes or are reported as spam or offensive. Figure 2 shows one sample of one question with its answers from StackOverflow. Till May 8, 2015, the statistics

Fig. 2: Illustration of one sample question from Stack Overflow.

of this site are as in Table. I.

B. Experiment Settings

In our experiment, part of StackOverflow dataset is used. We downloaded all questions posted from October 1, 2012 to December 31, 2012 and all related information like answers

number of users	4,232,639
number of votes	62,357,544
number of comments	44,557,809
number of questions	9,365,722
number of answers	15,632,696

TABLE I: The information of Stack Overflow till May 8, 2015.

was tracked until January 2014. This time period was chosen because of these reasons: First, questions and answers in this time period are not very out-dated; Second, few user activities on posts in this period are active. Thus, we assume that the best answer to one question is the final one. The dataset we use was dumped on January 2014 [4]. Before feature extraction, posts without users' IDs are removed. Then, only questions which have best answers and at least two more answers are considered. The final processed dataset has 52,104 questions and 190,165 answers. On average, there are 3.65 answers per question. During the experiments, our data set is randomly split into two parts evenly: training and testing.

To be specific, details as follows show how to generate relatively ranked pairs. For each question, only its best answer is considered as the high quality answer while others are treated as low-quality answers. Then each pair is generated in this way: one best answer and one of other answers to the same question. After all pairs are generated, feature extraction is performed based on information from three main aspects of each pair of questions and answers: content, interactions, users. These are briefly described below.

The First group of features are extracted based on the content of the answer in each pair of questions and answers. Part of these features are based on comments to the answers like *average score of comments, variance of the comments' scores, number of comments*. Comment-based features at least show that the corresponding answer is interesting and incur a good discussion towards problem solving. Besides these, whether one answer has pictures, URL or codes are also factors to show that the current answer has a high quality, since these components are able to show more information than text. Moreover, the length of answers [12][2] and its *readability* [18] also play an important role on answer quality.

Apart from the content information, features based on *interaction* are also considered, for example, the interaction between questions and answers, and that between different answers to one question. The first one is easy to understand since one answer has to be similar to its corresponding question, and thus the similarity between questions and answers is used as one feature. The second one is designed based on the assumption that users prefer the answers which is easy to understand. Computation of these features are shown in [18]. This is different from the feature interaction in our model. This one is on the feature-design level which focuses on exploring new information sources to design new features, while our case focuses on the model-design level.

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

User information also has an impact on the quality of answers. One answer is likely to have a high quality if the answerer is one expert. To represent the expertise of one user, these features are extracted based on users' previous activities, for example the number of answers one provides, how many questions one asks, the number of best answers he or she posts.

Our experiment is conducted by considering different groups of features and then results are presented respectively. In this way, it is easy to see the performance of different algorithms when we only consider informations from different aspects of our research problem (i.e. different groups of features). Finally, the experiment is conducted on the entire feature set we have. The three groups of features we consider in this experiment are: *content*, *interactions* and *user information*.

C. Experiment Results & Discussion

To show the performance of our proposed algorithm, we compare our model with approaches used in state-of-the-art. As mentioned in Section Introduction, there are two main trends in best answer prediction: one is to use classification techniques and then decision values are used as quality scores while the other one is to use ranking approaches directly. For the former case, linear support Vector Machine (SVM) is common used because data in social media is in large scale so that nonlinear algorithms are not computational efficient. In our experiment, linear SVM is the first baseline we choose. For the latter case, RankSVM [6] is used which is one main ranking algorithm used in the area of best answer prediction [5]. The code for RankSVM is from Microsoft Research [5]. On CQA sites, there are no direct information we can use as the metric to measure answer quality without manually labeling. For example, scores of each answer might be one proper metric. But this metric is not accurate. It is easy to see that it is easy for the answer which is posted early to have the high score. In fact, on Stack Overflow, there are a lot of answers having the higher scores than the corresponding best answers[6]. Thus in our experiments, we only treat the best answers as the high-quality ones and others as low-quality. As a result, in our experiment, it is the pairwise ranking problem so we do not compare with listwise ranking algorithms.

To make comparison between different models, two evaluation metrics are used: one is defined in Eqn. 26 and the other one is defined in Eqn. 27.

$$e_1 = \frac{\sum_{\forall(q_i, A_{i,j_1}, A_{i,j_2}) \in P} I(s_{i,j_1} > s_{i,j_2})}{|P|} \tag{26}$$

where s_{i,j_1}, s_{i,j_2} are predicted scores of A_{i,j_1}, A_{i,j_2} respectively. The relatively ranking set P is defined in Eqn. 2 and the function $I(\cdot)$ is shown in Eqn. 28.

[5]http://research.microsoft.com/en-us/um/beijing/projects/letor/baselines/ranksvm-primal.html

[6]https://data.stackexchange.com/stackoverflow/query/380215/where-accepted-answer-does-not-have-the-highest-score

$$g(i) = \arg\max_j \{s_{i,j}, j \in \{1, \cdots, M_i\}\}$$
$$e_2 = \frac{\sum_i I(j_{i,0} == g(i))}{N} \tag{27}$$

where $j_{i,0}$ is the index of the best answer of the i^{th} question, $s_{i,j}$ is the predicted score of the j^{th} answer of the i^{th} question and the function $g(\cdot)$ returns the index of the best answer of one given question and the function $I(\cdot)$ is given by Eqn.28.

$$I(x) = \begin{cases} 1 & \text{if } x \text{ is true} \\ 0 & \text{otherwise} \end{cases} \tag{28}$$

From the definitions, it is easy to see this fact: e_1 shows how good one algorithm is when it considers the pairwise ranking regardless of whether one algorithm can find the best answer to one question or not, while e_2 shows the performance of each algorithm when applied to best answer prediction. In other words, e_1 measures what percentage of relatively ranked pairs are predicted correctly, which focuses on the answer-level comparison. However e_2 measures what percentage of questions have the correctly predicted best answers.

To show the performance of different models on the pairwise ranking in best answer prediction, experiments were conducted to collect the metric e_1. The experimental results are shown in Table. II. Table. II presents the performance of algorithms used as learning to rank. From the results, we can see that our model performs best not only when only individual feature groups are considered but also when all features are considered. This shows that our model can be one good pairwise ranking algorithm in the area of community question and answering. From the results of SVM, we can see that when only f_c is considered, the performance is best. However, when simple concatenation of all features from different views is applied, the final one gives worse performance instead of better one. Similarly, for RankSVM, its performance is best when only f_u is considered. However after considering all features, the performance drops. For our approach, because we consider the interaction structure of features from different views, the final performance is best. This shows that there exists on latent interaction structure in the feature space. Incorporating weakly hierarchical lasso, we can capture this interaction structure. This shows the effectiveness of our proposed model.

	f_c	f_i	f_u	*all*
SVM	0.671	0.541	0.480	0.544
RankSVM	0.411	0.534	0.543	0.476
Ours	0.689	0.552	0.570	0.693

TABLE II: This table shows the results of different algorithms on Stack Overflow when considering the measurement metric e_1. Three groups of features: f_c content, f_i interactions, f_u user information.

To show comparison of performance on best answer prediction, experiments were run to collect metric e_2. Table. III presents the performance of different models. From the results, it is easy to see that our model performs best in the problem

	f_c	f_i	f_u	*all*
SVM	0.479	0.331	0.294	0.349
RankSVM	0.223	0.321	0.361	0.286
Ours	0.494	0.334	0.377	0.498

TABLE III: Experiment results (e_2) of different algorithms' performance. Three groups of features: f_c content, f_i interactions, f_u user information.

of best answer prediction not only when considering different groups of features independently but also when considering all features jointly. Similar to Table.II, the performance of SVM and RankSVM drop a lot when all features are considered by simple concatenation. For our model, it does not have this problem because of the fact that we incorporate the information from the latent interaction of features from different views.

Consequently, we conclude that the proposed models perform better than those in the state-of-the-art. Performance of experiments using both metrics shows the effectiveness of hierarchical interactions between different views in the problem of best answer prediction.

VI. Conclusion & Future Work

We present a new learning-to-rank approach to best answer prediction on CQA sites. Incorporating the weakly hierarchical lasso, our proposed model is able to effectively exploit the interactions of features from different views of the data. To find a solution under this new model, we reformulate it into one existing optimization framework. Experiments on Stack overflow are used to evaluate the proposed approach, with comparison to other methods in state-of-the-art. The experimental results demonstrate the effectiveness and superior performance of our approach. Although our algorithm is designed originally for best answer prediction, it can be treated as one ranking algorithm and used in most ranking situations. Thus the application of our algorithm in different areas can be one piece of future work. Moreover, in our algorithm, one limitation is that we study the interaction structure of different feature dimensions, instead of different groups of feature dimensions. Another interesting future work is to extending our algorithm by considering the hierarchical structure of different groups of feature dimensions.

Acknowledgment

This work was supported in part by a grant (#1135616) from National Science Foundation. Any opinions expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

References

[1] 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 *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2012, pp. 850–858.

[2] E. Agichtein, C. Castillo, D. Donato, A. Gionis, and G. Mishne, "Finding high-quality content in social media," in *Proceedings of the 2008 International Conference on Web Search and Data Mining*. ACM, 2008, pp. 183–194.

[3] C. Shah and J. Pomerantz, "Evaluating and predicting answer quality in community qa," in *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2010, pp. 411–418.

[4] D. H. Dalip, M. A. Gonçalves, M. Cristo, and P. Calado, "Exploiting user feedback to learn to rank answers in q&a forums: a case study with stack overflow," in *Proceedings of the 36th international ACM SIGIR conference on research and development in information retrieval*. ACM, 2013, pp. 543–552.

[5] Y. Cai and S. Chakravarthy, "Answer quality prediction in Q/A social networks by leveraging temporal features," *Proceedings of International Journal of Next-Generation Computing*, vol. 4, no. 1, 2013.

[6] O. Chapelle and S. S. Keerthi, "Efficient algorithms for ranking with svms," *Information Retrieval*, vol. 13, no. 3, pp. 201–215, 2010.

[7] F. Hieber and S. Riezler, "Improved answer ranking in social question-answering portals," in *Proceedings of the 3rd international workshop on Search and mining user-generated contents*. ACM, 2011, pp. 19–26.

[8] J. Jeon, W. B. Croft, J. H. Lee, and S. Park, "A framework to predict the quality of answers with non-textual features," in *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2006, pp. 228–235.

[9] B. Li, T. Jin, M. R. Lyu, I. King, and B. Mak, "Analyzing and predicting question quality in community question answering services," in *Proceedings of the 21st international conference companion on World Wide Web*. ACM, 2012, pp. 775–782.

[10] Y. Yao, H. Tong, T. Xie, L. Akoglu, F. Xu, and J. Lu, "Detecting high-quality posts in community question answering sites," *Information Sciences*, 2015.

[11] A. Shtok, G. Dror, Y. Maarek, and I. Szpektor, "Learning from the past: answering new questions with past answers," in *Proceedings of the 21st international conference on World Wide Web*. ACM, 2012, pp. 759–768.

[12] L. A. Adamic, J. Zhang, E. Bakshy, and M. S. Ackerman, "Knowledge sharing and yahoo answers: everyone knows something," in *Proceedings of the 17th international conference on World Wide Web*. ACM, 2008, pp. 665–674.

[13] M. Surdeanu, M. Ciaramita, and H. Zaragoza, "Learning to rank answers on large online qa collections." in *ACL*, 2008, pp. 719–727.

[14] ——, "Learning to rank answers to non-factoid questions from web collections," *Computational Linguistics*, vol. 37, no. 2, pp. 351–383, 2011.

[15] A. Agarwal, H. Raghavan, K. Subbian, P. Melville, R. D. Lawrence, D. C. Gondek, and J. Fan, "Learning to rank for robust question answering," in *Proceedings of the 21st ACM international conference on Information and knowledge management*. ACM, 2012, pp. 833–842.

[16] G. Burel, Y. He, and H. Alani, "Automatic identification of best answers in online enquiry communities," in *The Semantic Web: Research and Applications*. Springer, 2012, pp. 514–529.

[17] S. Ravi, B. Pang, V. Rastogi, and R. Kumar, "Great question! question quality in community q&a," in *Eighth International AAAI Conference on Weblogs and Social Media*, 2014.

[18] Q. Tian, P. Zhang, and B. Li, "Towards predicting the best answers in community-based question-answering services," in *Seventh International AAAI Conference on Weblogs and Social Media*, 2013.

[19] J. Bien, J. Taylor, R. Tibshirani *et al.*, "A lasso for hierarchical interactions," *The Annals of Statistics*, vol. 41, no. 3, pp. 1111–1141, 2013.

[20] Y. Liu, J. Wang, and J. Ye, "An efficient algorithm for weak hierarchical lasso," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014, pp. 283–292.

[21] P. Gong, C. Zhang, Z. Lu, J. Z. Huang, and J. Ye, "A general iterative shrinkage and thresholding algorithm for non-convex regularized optimization problems," in *Machine learning: proceedings of the International Conference. International Conference on Machine Learning*, vol. 28, no. 2. NIH Public Access, 2013, p. 37.

[22] J. Barzilai and J. M. Borwein, "Two-point step size gradient methods," *IMA Journal of Numerical Analysis*, vol. 8, no. 1, pp. 141–148, 1988.