Towards Personalized Activity Level Prediction in Community Question Answering Websites

Zhenguang Liu, Zhejiang University, and National University of Singapore
Yingjie Xia, Zhejiang University
Qi Liu, National University of Singapore
Qinming He, Zhejiang University
Yanxiang Chen, Hefei University of Technology
Roger Zimmermann, National University of Singapore

Community Question Answering (CQA) websites have become valuable knowledge repositories. Millions of internet users resort to CQA websites to seek answers to their encountered questions. CQA websites provide information far beyond a search on a site such as Google due to (1) the plethora of high quality answers, and (2) the capabilities to post new questions towards the communities of domain experts. While most research efforts have been made to identify experts or to preliminary detect potential experts of CQA websites, there has been a remarkable shift towards investigating how to keep the engagement of experts. Experts are usually the major contributors of high-quality answers and questions of CQA websites. Consequently, keeping the expert communities active is vital to improving the lifespan of these websites. In this paper, we present an algorithm termed PALP to predict the activity level of users of CQA websites. To the best of our knowledge, PALP is the first to address a personalized activity level prediction model for CQA websites. Furthermore, it takes into consideration user behavior change over time and focuses specifically on expert users. Extensive experiments on the Stack Overflow website demonstrate the competitiveness of PALP over existing methods.

CCS Concepts: • **Information systems → Decision support systems; Data analytics;**

General Terms: Algorithms, Prediction, Performance

Additional Key Words and Phrases: Question answering website, activity level prediction, logistic regression, personalized model

ACM Reference Format:
Zhenguang Liu, Yingjie Xia, Qi Liu, Qinming He, Yanxiang Chen, and Roger Zimmermann, 2016. Towards Personalized Activity Level Prediction in Community Question Answering Websites. *ACM Trans. Multimedia Comput. Commun. Appl.* 0, 0, Article 0 (0000), 15 pages.
DOI: 0000001.0000001

1. INTRODUCTION

Community Question Answering (CQA) websites have become valuable knowledge repositories in their specific domains [Pudipeddi et al. 2014] [Dror et al. 2012] [Nie et al. 2015]. Millions of internet users resort to CQA websites to seek answers to their encountered questions. CQA websites operate far beyond search engines such as

Author's addresses: Zhenguang Liu and Qi Liu and Roger Zimmermann, School of Computing, National University of Singapore, Singapore 119613; Yingjie Xia and Qinming He, Department of Computer Science, Zhejiang University, Hangzhou, China 310027; Yanxiang Chen, Hefei University of Technology, Hefei, China 230009.
Author's email: Zhenguang Liu liuzhenguang2008@gmail.com; Yingjie Xia xiayingjie@zju.edu.cn; Qi Liu qiliumize@gmail.com; Qinming He hqm@zju.edu.cn; Yanxiang Chen 872681166@qq.com; Roger Zimmermann rogerz@comp.nus.edu.sg.

DOI: 0000001.0000001

Google. (1) There are plenty of questions and high quality answers provided by CQA websites, where the user may find the same or a similar question to hers together with the associated answers. (2) Users can conveniently post their own questions to the crowdsourcing community full of experts. The expert community will provide professional answers in response to her questions. The main purpose of a CQA website is to produce and maintain high quality answers to users' questions. Voting, badges and reputation are typical mechanisms that guarantee the quality of questions and answers on these websites [Riahi et al. 2012]. As a successful example, Stack Overflow, a CQA website specializing in programming, has attracted 5,277,831 registered users and the number is still rapidly increasing.

However, CQA websites face their own problems and challenges [Pudipeddi et al. 2014]. One key problem is that these websites are required to continuously pay attention to the experts who have a tendency to churn among sites or become inactive. The popularity of a CQA website highly depends on the breadth of its questions and the quality of its answers. The departure of an expert implies the loss of a high quality content creator. As such, keeping the expert communities active is vital to improve the lifespan of these websites. Interestingly, only a small portion of users are responsible for answering a large portion of the questions [Riahi et al. 2012]. For example, in Stack Overflow the average number of posts (questions and answers) of a regular user is 6.88 while it is 167.11 for an expert. In other words, the majority of the users of a CQA website are content consumers who just browse the site, while only a limited number of experts are content producers who pose answers to questions.

A key question then is how to predict whether an expert will continue to be engaged with a site or is rather about to leave. If different experts have distinctive behavior patterns, can we develop a personalized prediction model for each expert? Furthermore, the behavior pattern of an expert may well change over time. For example, an expert may be very active in Stack Overflow when she is doing a project and may become inactive when the project is finished. How can we describe the dynamic nature of users in the process of prediction?

This paper seeks to answer the above questions. We present an algorithm termed PALP to predict the activity levels of users of CQA websites. Different from existing work, PALP addresses a personalized prediction model, takes into consideration the user behavior change over time, and focuses specifically on expert users. Since it is much easier to re-activate an expert who only shows signs of becoming inactive than when she has already left the site for a long time [Zhu et al. 2013] [Liu et al. 2016], expert activity level prediction can be very useful. First, the maintainer or owner of a site can leverage the information to identify experts with a high tendency to become inactive. Second, activity level prediction for all the experts allows for an early warning of an imminent period of intolerably few posts or preparations for a future post explosion.

We have conducted a case study on the Stack Overflow site, a typical CQA website specializing in programming related issues with over 5,277,831 users. The major contributions of this paper can be summarized as follows.

(1) To the best of our knowledge, the proposed PALP method is the first approach especially designed for personalized expert activity-level-prediction on CQA websites. Existing methods concentrate on presenting a general model to predict future activity levels of all users while ignoring the diversity of users and not focusing on experts.
(2) We validate the effectiveness of our proposed method by conducting a case study on a large-scale real world website, Stack Overflow. We not only compare PALP

with several baselines on the Stack Overflow data, but also construct and evaluate a rich variety of representative features for the prediction task.

(3) We take into consideration the dynamic nature of users in the process of prediction and propose a time decay function to penalize out-of-date training data.

The rest of this paper is organized as follows. In Section 2, we briefly review the related work. We then introduce the formal problem definition of expert activity level prediction in Section 3. Subsequently, we present a problem analysis and detail our proposed method in Section 4. Section 5 describes our extensive experiments and summarizes our findings. Finally, Section 6 concludes and elaborates on some insights for potential future work.

2. RELATED WORK

In the past decade, numerous interesting lines of work (e.g., [Zhu et al. 2013] [Grant and Betts 2013] [Pal 2015] [Liu and Huet 2016], [Wang et al. 2016]) studied CQA websites from different perspectives. The existing work can be roughly classified into three categories, namely high quality content mining (e.g., [Dalip et al. 2013] [Movshovitz-Attias et al. 2013]), expert finding and user churn prediction (e.g., [Pudipeddi et al. 2014] [Kawale et al. 2009] [Dasgupta et al. 2008] [Guyon et al. 2009] [Lampe and Johnston 2005]).

2.1. Content Mining

Content mining in CQA websites has a wide range of research directions but focuses mainly on mining high quality answers and detecting questions that are not sufficiently answered. For example, [Shah and Pomerantz 2010] designed various features from questions and answers and subsequently trained a classifier for selecting the best answer. They observed that the answerer's profile and the order of the answer in the list of answers are the two most significant features for predicting the best quality answers. [Dalip et al. 2013] formulated a random forest for ranking answers. [Anderson et al. 2012] regarded a question and its answers as an entity, and tried to identity the entities that will be of long lasting value. They also attempted to detect the questions that were not answered to the askers' satisfactory. In [Asaduzzaman et al. 2013], researches sought to answer why some questions remain unanswered and to predict the duration until the question would get its first answer. [Tian et al. 2013] tried to predict which answer will be the best answer for a question.

2.2. Expert Finding

Besides mining valuable content (questions and answers) in CQA sites, researchers have also put efforts into the identification of expert users.

Yue *et al.* established a prediction model to forecast whether or not a user will become an expert user from her initial behavior [Yue et al. 2012]. [Pal et al. 2011; Pal et al. 2012] discovered that it is possible to identify potential experts during the first two weeks of their joining a community. Early detection of potential experts can help the managers to nurture and retain these users. Pal *et al.* analyzed the behavioral patterns of experts over time and the interactions between experts. [Riahi et al. 2012] provided an approach to route new questions to the right group of experts such that experts are presented with questions matching their expertise. [Zhao et al. 2015] formulated the problem of expert finding into the problem of missing value estimation – they utilized graph regularized matrix completion to estimate missing values.

While most efforts were made in identifying experts or the preliminary detection of potential experts in CQA websites [Liu et al. 2013], recently there has been a remarkable shift towards activity level prediction. This stems from the fact that keeping the engagement of experts and trying to loose a minimal number of experts is crucial for the long-term sustainability of CQA websites.

2.3. User Churn Prediction

The most closely related work to activity level prediction is user churn prediction, which aims to identify users who will abandon the websites. [Dror et al. 2012] addressed the task of churn prediction for new users. They evaluated various features including personal information, activity rate and social interaction with others in the prediction. They concluded that the total number of answers and the positive responses such as upvotes are most relevant to user churn. However, they concentrated on new users rather than experts. Yang *et al.* [Yang et al. 2010] explored the new user survival patterns in three different question answering websites and revealed the diversity in users' participation lifespans in the three different sites. Similar to [Dror et al. 2012], [Yang et al. 2010] concentrated on new users.

In [Pudipeddi et al. 2014], the authors presented their study on churn prediction for both new and expert users. They constructed a long list of potentially indicative features and suggested that the time gaps between user posts can provide strong evidence for churn. However, they tried to train a general classifier for all the users and ignored their diverse personalities. [Zhu et al. 2013] developed a personalized model for user activity level prediction in social networks. However, the contexts in social networks are very different from that in CQA websites. A user in a social network can update his or her status, post photos, send messages and play various social games [Zhu et al. 2013]. In contrast, a user in a CQA website can only post questions and answers, and deliver an upvote or downvote to a post (i.e., either a question or an answer).

3. PROBLEM FORMULATION

We use the reputation in Stack Overflow to define experts. Namely, a user is an expert if and only if her reputation is higher than a threshold of 1,000 points. As such, we identified 13,542 experts out of all the 5,277,831 users on the site. The reputation of a user is proportional to her contributions on the website. For example, a user can earn 10 reputation points if her answer is voted up and can earn 15 reputation points if her answer is labeled as the accepted answer by the question asker. Note that for each question, there can only be one accepted answer among all the answers. A user can also lose reputation points when her answer is voted down or her posts receive 6 spam flags.

For simplicity, we formulate expert activity level prediction as a binary classification problem. An expert will be labeled as active (1) or inactive (0). However, the proposed framework can be easily extended to a c class version where the activity level of an expert is categorized into c different classes. After introducing the binary classification framework, we will further explain how to extend the framework to a multiple-class case. All the user activities are organized by month and the ground truth activity level is measured by their number of activities, which is the number of posts (a post is either a question or an answer), in a month. Now we formally define the problem of expert activity level prediction.

Definition 3.1 (*Expert activity level prediction problem*). For an expert, given her feature vector $x_i \in \mathbb{R}^d$ which is extracted from her past activities, predict his/her activity level $y_i \in \{1, 0\}$ in the next month, where 1 stands for active while 0 indicates inactive.

Further explanations for this definition are as follows. (1) The input feature vector x_i is extracted from the ith expert's past activities. For example, x_i may contain features such as the number of the user's posts in the last month, the number of the user's posts in the month before the last month, the total number of upvotes received by the user, the number of badges received in the last month, etc. (2) The goal is to predict the label $y_i \in \{0, 1\}$ for each expert x_i. We use the number of monthly posts to measure the activity level of an expert. This is based on the observation that posting questions and answers are the two most important activities in CQA sites. Note that a post is either a question or an answer. If an expert x_i does not submit any posts, she is labeled as inactive ($y_i = 0$), and active ($y_i = 1$) otherwise.

4. PROBLEM ANALYSIS AND THE PALP ALGORITHM

In Section 3, we have formulated the activity level prediction task into a binary classification problem. In this section, we will introduce a novel model to address the problem based on logistic regression.

Given an input feature $x \in \mathbb{R}^d$ (d is the dimensionality of the features), logistic regression predicts the target label of x utilizing the sigmoid function. Formally,

$$\hat{y} = h_\theta(x) = \frac{1}{1 + exp(-\theta^T x)} \tag{1}$$

where $\hat{y}$ is the activity level prediction of x, and x corresponds to the features extracted for an expert. $\theta = \langle \theta_1, \theta_2, \ldots, \theta_d \rangle^T$ is the parameter to be learned.

The base logistic regression model formulates the activity level prediction task into an optimization problem, i.e., finding the optimal solution θ that minimizes the cost function J. Formally, the problem can be formulated as follows.

$$\begin{aligned} \min_\theta J &= \frac{1}{n} \sum_{l=1}^{n} Cost(h_\theta(x_l^{(t)}), y_l^{(t+1)}) + \gamma_0 \|\theta\|_2^2 \\ &= -\frac{1}{n} \sum_{l=1}^{n} \{ y_l^{(t+1)} \, log(h_\theta(x_l^{(t)})) + (1 - y_l^{(t+1)}) \, log(1 - h_\theta(x_l^{(t)})) \} + \gamma_0 \|\theta\|_2^2 \end{aligned} \tag{2}$$

Four aspects of this formulation deserve further explanation. (1) n denotes the number of experts. (2) $x_l^{(t)}$ stands for the feature extracted for the lth expert at time t (corresponding to a month), where the features are extracted from her past activities up to time t. (3) $y_l^{(t+1)}$ is the ground truth activity level of expert x_l at time $t+1$, and $h_\theta(x_l^{(t)})$ is the activity level prediction. $h_\theta(x_l^{(t)}) = 1$ indicates that the expert x_l is predicted to be active at time $t+1$ while $h_\theta(x_l^{(t)}) = 0$ indicates inactivity. (4) The term $\|\theta\|_2^2$ is the regularization term to penalize the model complexity [Zhu et al. 2013].

The base model in Equation (2) can be trained to capture the optimal θ for the entire expert community. It is worth to emphasize that there exist plenty of training data for this model in the Stack Overflow website, which is primarily due to the large number of experts and the long existence of the site. However, this base model fails to reflect the diversity of the experts and capture the user behavioral changes over time. A personalized prediction model toward an expert will be more accurate than a general one, and the behavior pattern of an expert may change over time. Taking these aspects into consideration, we upgrade the base logistic regression model to one that is personalized with a time decay function as below.

4.1. User-specific Modeling

Different experts may have different behavior patterns. Using the same model to predict the activity level for different experts may be inaccurate for a specific expert. The base model in Equation (2) is trained to capture the optimal parameters for the entire expert community. As such, the base model can be regarded as capturing the common conduct of all the experts. A global model cannot produce precise predictions for a set of different users [Zhu et al. 2013]. Therefore, we propose a user-specific model for each expert by introducing a personalized term into Equation (2). In other words, besides the common factors, we further incorporate the historical data of a specific user to implement a more effective model for that user. Let w_i be the parameter to be trained for expert x_i. Then the model for expert x_i can be formulated as below.

$$\begin{aligned}\min_{\theta,w_i} J &= \frac{1}{n}\sum_{l=1}^{n} Cost(h_\theta(x_l^{(t)}), y_l^{(t+1)}) + \gamma_0\|\theta\|_2^2 + \alpha\,\frac{1}{t}\sum_{j=1}^{t} Cost(h_{w_i}(x_i^{(j)}), y_i^{(j+1)}) + \gamma_i\|w_i\|_2^2 \\ &= -\frac{1}{n}\sum_{l=1}^{n}\{y_l^{(t+1)}\, log(h_\theta(x_l^{(t)})) + (1-y_l^{(t+1)})\, log(1-h_\theta(x_l^{(t)}))\} + \gamma_0\|\theta\|_2^2 \\ &\quad -\alpha\,\frac{1}{t}\sum_{j=1}^{t}\{y_i^{(j+1)}\, log(h_{w_i}(x_i^{(j)})) + (1-y_i^{(j+1)})\, log(1-h_\theta(x_i^{(j)}))\} + \gamma_i\|w_i\|_2^2\end{aligned} \tag{3}$$

The global parameter θ captures the global knowledge across the expert community and the user-specific parameter w_i captures the personalized pattern of user x_i. The first term on the right side of the equation is the global term, while the third term is the personalized term. The second term $\gamma_0\|\theta\|_2^2$ and the fourth term $\gamma_i\|w_i\|_2^2$ are the regularization terms. Note that in the third term, analogous to Equation (1),

$$h_{w_i}(x) = \frac{1}{1+exp(-{w_i}^T x)} \tag{4}$$

The parameter α in Equation (3) controls the tradeoff between commonality across experts and the personalization [Zhu et al. 2013]. For example, a relatively large α implies an emphasis of the influence of personalization. Noticeably, a too large value of α may lead to overfitting since the historical data set of an expert is usually small.

4.2. Time Decay Setting

A user's behavior may well change over time. For example, an expert may actively post new questions and answers on the Stack Overflow website when she is doing a programming project. But when her project ends, she may become inactive on the site. Simultaneously, an expert may lose interest in her past topics and become active on a new topic. Obviously, a user's behavior pattern is more similar to her recent activities rather than her activities a long time before. Thus, it is necessary to increase the weight of recent data and penalize the out-of-date training data. Mathematically, we introduce a time decay setting in the model of Equation (3) as below.

$$\begin{aligned}\min_{\theta,w_i} J &= \frac{1}{n}\sum_{l=1}^{n} Cost(h_\theta(x_l^{(t)}), y_l^{(t+1)}) + \gamma_0\|\theta\|_2^2 + \alpha\,\frac{1}{t}\sum_{j=1}^{t} Cost(h_{w_i}(x_i^{(j)}), y_i^{(j+1)}) + \gamma_i\|w_i\|_2^2 \\ &= -\frac{1}{n}\sum_{l=1}^{n}\{y_l^{(t+1)}\, log(h_\theta(x_l^{(t)})) + (1-y_l^{(t+1)})\, log(1-h_\theta(x_l^{(t)}))\} + \gamma_0\|\theta\|_2^2 \\ &\quad -\alpha\,\frac{1}{t}\sum_{j=1}^{t} e^{-\beta(t-j)}\{y_i^{(j+1)}\, log(h_{w_i}(x_i^{(j)})) + (1-y_i^{(j+1)})\, log(1-h_\theta(x_i^{(j)}))\} + \gamma_i\|w_i\|_2^2\end{aligned} \tag{5}$$

ALGORITHM 1: The PALP Algorithm

Input: The set of features $\{x_i^{(j)} | i = 1, 2, \ldots, n; \;\; j = 1, 2, \ldots, t\}$ where $x_i^{(j)}$ is the feature representation of the ith expert in month j; tradeoff parameter α; time decay rate β; regularization parameters γ_0, γ_i; learning rate η

Output: The activity level prediction $\hat{y_i}^{(t+1)}$ of each expert for the next month $t+1$

Generate the initial value of θ and w_i $(i = 1, 2, \ldots, n)$ randomly;
iter_num = 0;
repeat
  Fix all w_i, keep updating θ by: $\theta = \theta - \eta \frac{\partial J}{\partial \theta}$, where $\frac{\partial J}{\partial \theta}$ is defined in Equation (6);
  for $i = 1 : n$ **do**
    Fix θ, keep updating w_i by: $w_i = w_i - \eta \frac{\partial J}{\partial w_i}$, where $\frac{\partial J}{\partial w_i}$ is defined in Equation (6);
  end
  iter_num = iter_num+1;
until *convergence* **or** *iter_num* $\geqslant$ *iter_threshold*;
for $i = 1 : n$ **do**
  $\hat{y_i}^{(t+1)} = h_\theta(x_i^{(t)}) + \alpha h_{w_i}(x_i^{(t)})$;
end

In Equation (5), the term $e^{-\beta(t-j)}$ is added to control the weight of the expert's historical data and β is termed the time decay rate. It is noteworthy that more recent historical data has a higher weight.

4.3. The PALP Algorithm

In summary, our goal is to solve the optimization problem in Equation (5). There are two parameters to be learned, namely θ and w_i. The gradient descents of cost J with respect to θ and w_i are

$$\begin{aligned}
\frac{\partial J}{\partial \theta} &= -\frac{1}{n}\sum_{l=1}^{n}(h_\theta(x_l^{(t)}) - y_l^{(t+1)})x_l^{(t)} + 2\gamma_0\theta \\
\frac{\partial J}{\partial w_i} &= -\alpha\,\frac{1}{t}\sum_{j=1}^{t} e^{-\beta(t-j)}(h_{w_i}(x_i^{(j)}) - y_i^{(j+1)})x_i^{(j)} + 2\gamma_i w_i
\end{aligned} \tag{6}$$

Given the derivatives above, we can update θ and all w_i $(i = 1, 2, \ldots, n)$ in an alternating way. In other words, during each iteration, we first fix all w_i and update θ, then fix θ and update all w_i. The updates of θ and w_i $(i = 1, 2, \ldots, n)$ follow the gradient descent method as below

$$\begin{aligned}
\theta &= \theta - \eta\frac{\partial J}{\partial \theta} \\
w_i &= w_i - \eta\frac{\partial J}{\partial w_i}
\end{aligned} \tag{7}$$

where η is the learning rate.

The overall algorithm termed PALP is illustrated in Algorithm 1. At the very beginning, we randomly initialize the value of θ and w_i $(i = 1, 2, \ldots, n)$. Afterwards, we use the gradient descent method to find the optimal θ and w_i $(i = 1, 2, \ldots, n)$. We first fix all w_i, and keep updating θ by $\theta = \theta - \eta\frac{\partial J}{\partial \theta}$. Second, we fix θ, and keep updating w_i by $w_i = w_i - \eta\frac{\partial J}{\partial w_i}$. Finally, the model is trained and the optimal θ and w_i of the model are obtained. We leverage the trained model to predict the activity level of each expert for the next month.

More specifically, the first for-loop in Algorithm 1 is to capture the personalized w_i for each expert, while the second for-loop predicts the activity level of each expert. Since the training of two different experts are uncorrelated, the operations in the first for-loop can be fully parallelized such that the algorithm can be accelerated greatly. Similarly, the operations in the second for-loop can also be parallelized.

4.4. Extension to Multi-Class Case

In the above problem analysis of this section (Section 4), we formulated the expert activity level prediction as a binary classification problem. Each expert will be labeled as either active (1) or inactive (0). In this subsection, we discuss how to extend the binary classification framework to a multi-class case.

The extension is quite straight forward. We adopt the one-vs-all or one-vs-rest method. That is, for each class m, we regard m as a class and regard the remaining classes as the other one class. Then we can simply utilize the above binary classification framework to train a model for class m. (1) Each class m is trained to have one classifier, which is deployed to predict the probability of label $y(x) = m$ for an input feature vector x. (2) We input x into the classifiers for each class and subsequently calculate the probability of $y(x) = 1, 2, \ldots, c$, where c is the number of classes, respectively. We consider the class of the maximum probability as the label of x.

5. EXPERIMENTS

In this section, we conduct extensive experiments to evaluate the PALP algorithm with a case study on the Stack Overflow website. First, we discuss the construction of features from Stack Overflow. Then, we evaluate our proposed PALP method against three baseline methods. Afterwards, we study the sensitivity of PALP with respect to its parameters. Finally, we explore the influence of different features by abandoning part of the features and re-running the experiments, respectively.

5.1. Feature Construction

In Stack Overflow, there exists a rich set of activities including posting a question, posting an answer, suggesting the editing of a post, voting up a post (i.e., a question or an answer), and voting down a post. Therefore, various features for users can be extracted from these activities. We construct and evaluate an extensive list of features that may be indicative for the activity level prediction task. We also classify all the constructed features into three categories.

As listed in Table I, we construct features and classify them into three groups. The first group is post related, where a post is either a question or an answer. The second group is positive response related. The third group is the profile of the expert. In the table, k denotes the month where the activity level is to be predicted. In the process of the prediction, the activities of the expert in the past T months, i.e., months $k-1, k-2, \ldots, k-T$, are extracted. Thus, T is the width of the sliding time window. We further introduce the features listed in Table I below.

(1) Feature *num_posts*$(k-i)$ captures the number of posts in the month that is i months ago. The number of posts during the $(k-i)^{th}$ month may well be a potentially indicative feature since we are aiming to predict the number of posts of the k^{th} month.
(2) Feature *num_questions*$(k-i)$ captures the number of questions in the month that is i months ago. The number of questions of the $(k-i)^{th}$ month may well be a potentially indicative feature since a post is either a question or an answer.

Table I. List of Features Constructed

Feature	Description
num_posts($k-1$)	Number of posts in the $(k-1)^{th}$ month.
num_posts($k-2$)	Number of posts in the $(k-2)^{th}$ month.
...	...
num_posts($k-T$)	Number of questions in the $(k-T)^{th}$ month.
num_questions($k-1$)	Number of questions in the $(k-1)^{th}$ month.
num_questions($k-2$)	Number of questions in the $(k-2)^{th}$ month.
...	...
num_questions($k-T$)	Number of questions in the $(k-T)^{th}$ month.
num_answers($k-1$)	Number of answers in the $(k-1)^{th}$ month.
num_answers($k-2$)	Number of answers in the $(k-2)^{th}$ month.
...	...
num_answers($k-T$)	Number of answers in the $(k-T)^{th}$ month.
post_length($k-1$)	Average length of posts in the $(k-1)^{th}$ month.
post_length($k-2$)	Average length of posts in the $(k-2)^{th}$ month.
...	...
post_length($k-T$)	Average length of posts in the $(k-T)^{th}$ month.
num_accepted_answers($k-1$)	Number of accepted answers in the $(k-1)^{th}$ month.
num_accepted_answers($k-2$)	Number of accepted answers in the $(k-2)^{th}$ month.
...	...
num_accepted_answers($k-T$)	Number of accepted answers in the $(k-T)^{th}$ month.
num_badges($k-1$)	Number of badges in the $(k-1)^{th}$ month.
num_badges($k-2$)	Number of badges in the $(k-2)^{th}$ month.
...	...
num_badges($k-T$)	Number of badges in the $(k-T)^{th}$ month.
reputation	The reputation of the expert.
total_num_questions	The total number of questions posted by the expert.
total_num_answers	The total number of answers posted by the expert.
total_upvotes	The total upvotes received by the expert.
total_downvotes	The total downvotes received by the expert.
total_badges	The total badges received by the expert.

(3) Feature *num_answers*$(k-i)$ captures the number of answers in the month that is i months ago.
(4) Feature *post_length* captures the average length of a user's posts in the month that is i months ago. A longer length of posts usually implies that the expert is more interested in the website.
(5) Feature *num_accepted_answers*$(k-i)$ captures how many accepted answers the expert received in the month that is i months ago. The insight is that the number of accepted answers signals how much the peer community admits the quality of the expert's answers and thus may have connections with the expert activity level.
(6) Feature *num_badges*$(k-i)$ captures the number of badges received in the month that is i months ago. A badge can only be awarded to a user who is especially helpful. Winning a badge may incentivize experts to produce more high-quality content and be more engaged with the website.
(7) Feature *reputation* is the most important item in the user profile. The reputation of a user is directly proportional to her contributions to the site. For example, a user can earn 10 reputation points if one of her answers receives an upvote and conversely can lose 2 reputation points if her answer receives a downvote.

(8) Features *total_num_questions* and *total_num_answers* are the total number of questions and answers the expert has posted on the website, respectively. The total number of questions and answers may signal the attitude of the expert towards the website.
(9) Features *total_upvotes* and *total_downvotes* are the total number of upvotes and downvotes the expert received on the website, respectively. The total number of upvotes and downvotes signals the agreement and disagreement votes the expert received from the peer community, which may affect the attitude of the expert towards the website.
(10) Feature *total_badges* is the total number of badges received by the expert.

Rather than considering all the user activities right from the very beginning, we only consider the activities within $T = 4$ months from the current month. In other words, the width of the sliding time window is set to $T = 4$ in our experiments. There are two reasons for this selection. First, we have observed that the activities that are 5 months or longer ago have little connection with the user's activity level 5 months later. Second, the computational costs can be substantially reduced via this selection.

5.2. Performance Evaluation

For all the 13,542 experts on Stack Overflow, we extracted their features for the first seven months of 2015 according to Table I. The activity levels of the experts during each month from May to August 2015 are also labeled according to the ground truth. Specifically, if an expert has no less than one post during the month, she is labeled as active for the month and otherwise labeled as inactive. We utilized all the data of year 2014 to construct the training data set. We leveraged the extracted activity features in every four successive months as input features, and the activity level of the next month as the labels, to create the training data set.

For comparison, we selected three types of baseline methods. Those are the Base Logistic Regression (**BLR**) model, the SVM classifier (**SVM**) and the K-Nearest Neighbor classifier (**KNN**). The BLR model is the base logistic regression model that ignores personalization and the user behavior change over time. For the SVM classifier, we utilize a Gaussian radial basis function as the kernel to enable a non-linear classifier. For the KNN classifier, we vary the number K of nearest neighbors from 1 to 30 with an interval of 5 and only report the result of its highest accuracy, where accuracy is the ratio of precisely predicted experts out of all experts throughout the paper.

Table II. Performance Comparison of Different Methods In Terms of Accuracy.

Method	BLR	SVM	KNN	PALP
May 2015	0.6949	0.6746	0.6187	**0.7778**
June 2015	0.6693	0.6858	0.5195	**0.7319**
July 2015	0.6718	0.6441	0.5806	**0.7255**
August 2015	0.6546	0.6822	0.6087	**0.7283**

Table II shows the performance comparison between the four methods. We observe that KNN performs the worst among all the four methods. This implies that the Euclidean distance between the user features may not be a good measure for the similarity between users. We can also see that our proposed PALP method consistently outperforms the three other methods in terms of prediction accuracy. BLR has comparable performance as SVM on the data set. Since PALP, BLR and SVM are all trained on the same data set (all the data of the experts of year 2014), this further verifies the observation that a personalized prediction model for an expert will be more accurate than a general one, and considering the user-behavior-pattern change over time can be helpful in prediction.

Our next experiments compare PALP with two reductions of PALP. One of the reductions is termed PALP_NoDecay where the time decay setting is reduced from PALP. The other reduction is the base logistic regression BLR introduced before. In the BLR method, all the personalizations are ignored. The comparison results are shown in Table III. We can clearly observe that BLR performs the worst among all the three methods. PALP_NoDecay performs better than BLR but still not as well as PALP. PALP consistently outperforms its two reductions. The results suggest that both, personalization modeling and time decay setting, lead to an improvement in prediction accuracy. Another finding from the results is that personalization modeling seems to be slightly more important in prediction than the time decay setting.

Table III. Performance Comparison with Two Reductions In Terms of Accuracy.

Method	BLR	PALP_ NoDecay	PALP
May 2015	0.6949	0.7201	**0.7778**
June 2015	0.6693	0.6942	**0.7319**
July 2015	0.6718	0.6911	**0.7255**
August 2015	0.6546	0.6880	**0.7283**

5.3. Robustness Study

In this subsection we study the effects of several parameters on the performance of the PALP algorithm. There are three key parameters to be learned: (1) the tradeoff parameter α, which controls the balance between the commonality across experts and the personalization, (2) the time decay rate β, and (3) the regularization parameter γ_i.

In order to study the effect of the tradeoff parameter α, we fixed all the other parameters and varied α from 0 to 10. Figure 1 shows the performance evolution with respect to parameter α. The figure suggests that small values of α yield better performance than bigger values. $\alpha = 0.2$ yields the best performance. The reasons are that the training data for one expert is usually sparse while the training data for the entire expert community is rich, so the weight of the personalization should be relatively small to avoid overfitting.

Next, we investigated the effect of the time decay rate β. We varied the value of β from 0 to 5. The performance evolution with respect to parameter β is demonstrated in Figure 2. We can clearly see that when the value of β is not greater than 0.5, PALP performs well. When the value of β equals 0.3, the performance of PALP reaches its peak. A too big value of β, e.g., bigger than 1, yields a poor performance of PALP. This may stem from the fact that a too large β will lead to over penalization of the data that is a few months ago [Zhu et al. 2013].

Finally, we evaluated the effect of the regularization parameter γ_i. We fixed the global regularization parameter $\gamma_0 = 1$ and varied γ_i from 0 to 10. The performance evolution with respect to parameter γ_i is shown in Figure 3. We can observe that parameter setting $\gamma_i = 1$ reaches the best performance. Relatively bigger values of γ_i have better performance than too small ones.

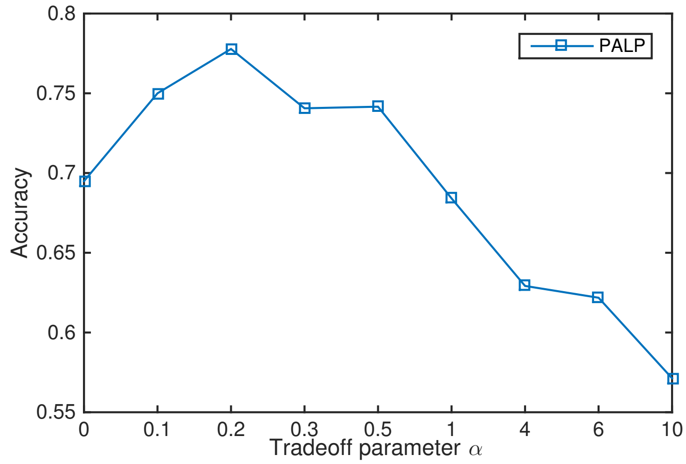

Fig. 1. Performance evolution with respect to tradeoff parameter α.

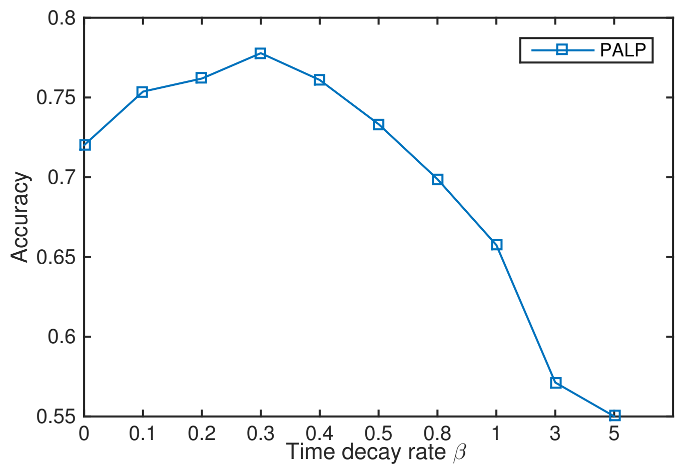

Fig. 2. Performance evolution with respect to time decay parameter β.

5.4. Analysis on Reduced Features

We also conducted experiments to study the effects of abandoning some features. We classified all the constructed features listed in Table I into three categories: post related, positive response related, and user profile related. We removed each of the three categories from the feature set and then tested the performance, respectively.

The comparison results are illustrated in Table IV. We can observe that the post related features, such as the monthly number of posts, questions and answers, have the most dominant effect on the accuracy of the prediction task. Positive response related features, such as the monthly number of accepted answers and the number of badges have the second most dominant effect right after the post related features. Surprisingly, we find that the user profile features, such as reputation and the total number of badges obtained by the expert, are not so informative in predicting the future activity level of the expert. The reasons may be that features extracted from

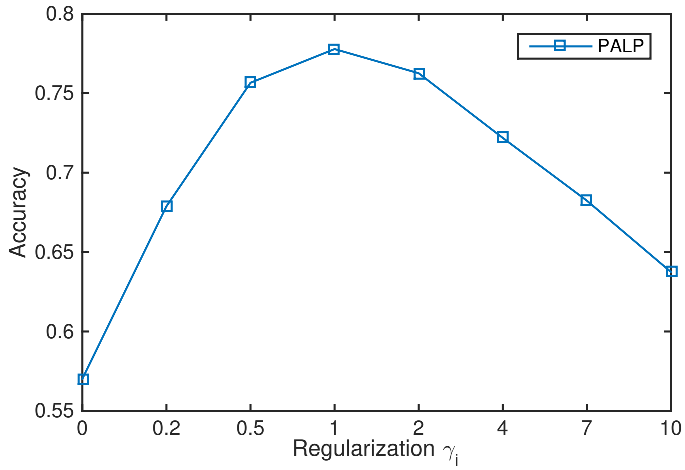

Fig. 3. Performance evolution with respect to parameter γ_i.

Table IV. Performance Comparison with Reduced Features.

All Features	0.7778
No Post Related Features	0.5943
No Positive Response Related Features	0.6839
No User Profile Features	0.7401

an expert's recent activities are more indicative than the features extracted from the expert's activities counting from the very beginning [Liu et al. 2016]. We can also see that combining all the features indeed leads to an improvement in prediction accuracy, which suggests the complementary nature of the different features.

6. CONCLUSIONS

In this paper we have proposed a method termed PALP to address the personalized activity level prediction for expert users in CQA websites. We have also taken into consideration the behavior change of an expert over time to enable more accurate prediction. Further, we have constructed and evaluated an extensive list of potential indicative features and classified them into three categories. Experimental results demonstrate the superior performance of our approach over the existing methods. We have also observed that the post related features, and the positive response related features, are the two most indicative factors for the activity level prediction task. As possible future work, we will test our model on CQA websites of different countries to study how the cultural differences may affect the model. We will also incorporate other information channels to further improve the accuracy of the prediction since various mobile apps and social networks may capture much more information of the user.

REFERENCES

Ashton Anderson, Daniel P. Huttenlocher, Jon M. Kleinberg, and Jure Leskovec. 2012. Discovering value from community activity on focused question answering sites: a case study of stack overflow. In *The 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '12*. Beijing, China, 850–858.

Muhammad Asaduzzaman, Ahmed Shah Mashiyat, Chanchal K Roy, and Kevin A Schneider. 2013. Answering questions about unanswered questions of stack overflow. In *Proceedings of the 10th Working Conference on Mining Software Repositories*. Santiago, Chile, 97–100.

Daniel Hasan Dalip, Marco Cristo, and Pavel Calado. 2013. Exploiting User Feedback to Learn to Rank Answers in Q&A Forums: a Case Study with Stack Overflow. In *Proceedings of SIGIR 13*. Dublin, Ireland, 543–552.

Koustuv Dasgupta, Rahul Singh, Balaji Viswanathan, Dipanjan Chakraborty, Sougata Mukherjea, Amit A Nanavati, and Anupam Joshi. 2008. Social ties and their relevance to churn in mobile telecom networks. In *Proceedings of the 11th international conference on Extending database technology: Advances in database technology*. 668–677.

Gideon Dror, Dan Pelleg, Oleg Rokhlenko, and Idan Szpektor. 2012. Churn prediction in new users of Yahoo! answers. In *Proceedings of the 21st international conference companion on World Wide Web*. 829–834.

Scott Grant and Buddy Betts. 2013. Encouraging user behaviour with achievements: an empirical study. In *Proceedings of the 10th Working Conference on Mining Software Repositories, MSR '13*. San Francisco, CA, USA, 65–68.

Isabelle Guyon, Vincent Lemaire, Marc Boullé, Gideon Dror, and David Vogel. 2009. Analysis of the KDD Cup 2009: Fast Scoring on a Large Orange Customer Database. In *KDD Cup*. 1–22.

Jaya Kawale, Aditya Pal, and Jaideep Srivastava. 2009. Churn prediction in MMORPGs: A social influence based approach. In *Computational Science and Engineering, 2009. CSE'09. International Conference on*. 423–428.

Cliff Lampe and Erik Johnston. 2005. Follow the (slash) dot: effects of feedback on new members in an online community. In *Proceedings of the 2005 international ACM SIGGROUP conference on Supporting group work*. 11–20.

Duen Ren Liu, Yu Hsuan Chen, Wei Chen Kao, and Hsiu-Wen Wang. 2013. Integrating expert profile, reputation and link analysis for expert finding in question-answering websites. *Information Processing & Management* 49, 1 (2013), 312–329.

Xueliang Liu and Benoit Huet. 2016. Event-based cross media question answering. *Multimedia Tools and Applications* 75, 3 (2016), 1495–1508.

Zhenguang Liu, Qi Liu, Beibei Zhang, Yifang Yin, Shubham Goyal, and Roger Zimmermann. 2016. Expert Activity Level Prediction on Question Answering Sites: A Case Study on StackOverflow. (2016). Accepted.

Dana Movshovitz-Attias, Yair Movshovitz-Attias, Peter Steenkiste, and Christos Faloutsos. 2013. Analysis of the reputation system and user contributions on a question answering website: Stackoverflow. In *2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. 886–893.

Liqiang Nie, Meng Wang, Luming Zhang, Shuicheng Yan, Bo Zhang, and Tat-Seng Chua. 2015. Disease Inference from Health-Related Questions via Sparse Deep Learning. *IEEE Trans. Knowl. Data Eng* 27, 8 (2015), 2107–2119.

Aditya Pal. 2015. Discovering Experts across Multiple Domains. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. Santiago, Chile, 923–926.

Aditya Pal, Shuo Chang, and Joseph A. Konstan. 2012. Evolution of Experts in Question Answering Communities. In *Proceedings of the Sixth International Conference on Weblogs and Social Media*. Dublin, Ireland, 274–281.

Aditya Pal, Rosta Farzan, Joseph A. Konstan, and Robert E. Kraut. 2011. Early detection of potential experts in question answering communities. In *Proceedings of the 19th conference on User Modeling, Adaption and Personalization*. 231–242.

Jagat Sastry Pudipeddi, Leman Akoglu, and Hanghang Tong. 2014. User churn in focused question answering sites: characterizations and prediction. In *23rd International World Wide Web Conference, WWW '14, Companion Volume*. Seoul, Republic of Korea, 469–474.

Fatemeh Riahi, Zainab Zolaktaf, Mahdi Shafiei, and Evangelos Milios. 2012. Finding expert users in community question answering. In *Proceedings of the 21st International Conference on World Wide Web*. 791–798.

Chirag Shah and Jefferey Pomerantz. 2010. Discovering value from community activity on focused question answering sites: a case study of stack overflow. In *Proceedings of SIGIR 10*. Geneva, Switzerland., 411–418.

Qiongjie Tian, Peng Zhang, and Baoxin Li. 2013. Towards Predicting the Best Answers in Community-based Question-Answering Services. In *ICWSM*.

Wei Wang, Yan Yan, Luming Zhang, Richang Hong, and Nicu Sebe. 2016. Collaborative Sparse Coding for Multi-view Action Recognition. *IEEE Multimedia Magazine* 23, 4 (2016), 80–87.

Jiang Yang, Xiao Wei, Mark S. Ackerman, and Lada A. Adamic. 2010. Activity Lifespan: An Analysis of User Survival Patterns in Online Knowledge Sharing Communities. In *Proceedings of the Fourth International Conference on Weblogs and Social Media, ICWSM 2010*. Washington, DC, USA.

Cheng Yue, Royce, Richard Hsu, and Nicholas Stevens. 2012. Discovering Elite Users in Question and Answering Communities. *CS* (2012), 1–5.

Zhou Zhao, Xiaofei He Lijun Zhang, and Wilfred Ng. 2015. Expert Finding for Question Answering via Graph Regularized Matrix Completion. *IEEE Transactions on Knowledge and Data Engineering* 27, 4 (2015), 1–12.

Yin Zhu, ErHeng Zhong, Sinno Jialin Pan, Xiao Wang, Minzhe Zhou, and Qiang Yang. 2013. Predicting user activity level in social networks. In *22nd ACM International Conference on Information and Knowledge Management, CIKM'13*. San Francisco, CA, USA, 159–168.

Received October 2016; revised XXX XXX; accepted XXX XXX