Popularity Modeling for Mobile Apps: A Sequential Approach

Hengshu Zhu, Chuanren Liu, Yong Ge, Hui Xiong, *Senior Member, IEEE,* and Enhong Chen, *Senior Member, IEEE*

***Abstract*—The popularity information in App stores, such as chart rankings, user ratings, and user reviews, provides an unprecedented opportunity to understand user experiences with mobile Apps, learn the process of adoption of mobile Apps, and thus enables better mobile App services. While the importance of popularity information is well recognized in the literature, the use of the popularity information for mobile App services is still fragmented and under-explored. To this end, in this paper, we propose a sequential approach based on hidden Markov model (HMM) for modeling the popularity information of mobile Apps toward mobile App services. Specifically, we first propose a popularity based HMM (PHMM) to model the sequences of the heterogeneous popularity observations of mobile Apps. Then, we introduce a bipartite based method to precluster the popularity observations. This can help to learn the parameters and initial values of the PHMM efficiently. Furthermore, we demonstrate that the PHMM is a general model and can be applicable for various mobile App services, such as trend based App recommendation, rating and review spam detection, and ranking fraud detection. Finally, we validate our approach on two real-world data sets collected from the Apple Appstore. Experimental results clearly validate both the effectiveness and efficiency of the proposed popularity modeling approach.**

***Index Terms*—App recommendation, hidden Markov models (HMMs), mobile Apps, popularity modeling.**

I. Introduction

WITH the rapid development of mobile App industry, the number of mobile Apps available has exploded over the past few years. For example, as of the end of April 2013, there are more than 1.6 million Apps at Apple Appstore [2] and Google Play [1]. To facilitate the adoption of mobile Apps and understand the user experience with mobile Apps, many App stores provide the periodical (such as daily) App chart rankings and allow users to post ratings and reviews for their Apps. Indeed, such popularity information plays an important role in mobile App services [9], [25], [30], and opens a venue for mobile App understanding, trend analysis, and other related applications [18], [27].

While people have developed some specific approaches to explore the popularity information of mobile Apps for some particular tasks [30], [31], [33], the use of popularity information for mobile App services is still fragmented and under-researched. Indeed, there are two major challenges along this line. First, the popularity information of mobile Apps often varies frequently and has the instinct of sequence dependence. For example, although the daily rankings of different mobile Apps may be different, it is unlikely that an App with a high ranking will be ranked very low in the following day due to the momentum of popularity. Second, the popularity information is heterogeneous, but contains latent semantics and relationships. For example, while ranking = 1 and rating = 5 come from different observations, both of them can indicate the high popularity.

To this end, in this paper, we propose a sequential approach based on hidden Markov model (HMM) to model the heterogeneous popularity information of mobile Apps. Particularly, this paper is to provide a comprehensive modeling of popularity information for mobile App services. Along this line, we first propose a popularity based HMM (PHMM) by extending the original HMM with heterogeneous popularity information of mobile Apps, including chart rankings, user ratings, and review topics extracted from user reviews. In this paper, the popularity information can be modeled in terms of different transitions of different popularity states, which indicate the latent semantics and relationships of popularity observations. Then, to efficiently train the proposed PHMM, we introduce a bipartite based method to precluster various popularity observations. The preclustering results can be leveraged for choosing parameters and assigning the initial values of PHMM. Indeed, the proposed PHMM is a general model, and many novel applications may benefit from the results of training PHMM. To show this, in this paper, we further introduce how to leverage PHMM for popularity prediction, and have a focus on demonstrating some novel mobile App services enabled by PHMM, including trend based App

Manuscript received November 26, 2013; revised May 14, 2014 and August 4, 2014; accepted August 5, 2014. Date of publication September 4, 2014; date of current version June 12, 2015. This work was supported in part by the National Science Foundation for Distinguished Young Scholars of China under Grant 61325010, in part by the Natural Science Foundation of China under Grant 71329201, in part by the National High Technology Research and Development Program of China under Grant 2014AA015203, in part by the National Science Foundation under Grant CCF-1018151 and Grant IIS-1256016, and in part by UNC Charlotte Faculty Research Grant 2014-2015. This paper was recommended by Associate Editor J. Liu.

H. Zhu and E. Chen are with the School of Computer Science and Technology, University of Science and Technology of China, Hefei 230026, China (e-mail: zhs@mail.ustc.edu.cn; cheneh@ustc.edu.cn).

C. Liu and H. Xiong are with the Management Science and Information Systems Department, Rutgers Business School, Rutgers University, Newark, NJ 07102 USA (e-mail: chuanren.liu@rutgers.edu; hxiong@rutgers.edu).

Y. Ge is with the Department of Computer Science, College of Computing and Informatics, University of North Carolina at Charlotte, Charlotte, NC 28223 USA (e-mail: yong.ge@uncc.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2014.2349954

recommendation, rating and review spam detection, and ranking fraud detection for mobile Apps. To be more specific, the contributions of this paper can be summarized as follows.

1) First, to our best knowledge, this paper provides the first comprehensive study of popularity modeling for mobile Apps. Indeed, this paper presents a simple, but practical way for exploring the popularity information of mobile Apps, and thus it is critically important for the successful development of various mobile App services.
2) Second, we propose a novel sequential approach, for solving the problem of popularity modeling for mobile Apps. Specifically, this approach contains a sequential model named PHMM that is extended from the original HMM for modeling the heterogeneous popularity observations of Apps, and a bipartite graph based method for parameter estimation and initial value assignment.
3) Third, we demonstrate many novel mobile App services enabled by PHMM, including trend based App recommendation, rating and review spam detection, and ranking fraud detection.
4) Finally, to validate the proposed popularity modeling approach, we conduct extensive experiments on two real-world data sets collected from the Apple Appstore, which cover a long time period. Experimental results clearly demonstrate both the effectiveness and efficiency of the proposed approach.

Overview: The remainder of this paper is organized as follows. In Section II, we briefly introduce the related works of this paper. In Section III, we introduce some preliminaries and give the problem statement. Section IV shows the details of our HMM based popularity modeling approach. In Section V, we provide some discussions about the applications of the proposed PHMM. Section VI presents the experimental results. Finally, Section VII concludes the work.

II. Related Work

In general, the related works of this paper can be grouped into two categories.

The first category is about the mobile App services. For instance, Yan and Chen [30] developed a mobile App recommender system, named Appjoy, which is based on users' App usage records to build a preference matrix instead of using explicit user ratings. To solve the sparsity problem of App usage records, Shi and Ali [25] studied several recommendation models and proposed a content based collaborative filtering model, named Eigenapp, for recommending Apps in their web site Getjar. Also, some researchers studied the problem of exploiting enriched contextual information for mobile App recommendation. For example, Zhu *et al.* [32] proposed a uniform framework for personalized context-aware recommendation, which can take both context independency and dependency assumptions into consideration. The framework can mine user's personal context-aware preferences from the context logs of mobile users for mobile App recommendation. In addition, Liu *et al.* [19] presented a survey on mobile context-aware recommendation, which introduces many novel mobile App services.

Meanwhile, the problem of detecting the rating and review spam is also important for App services. For example, Lim *et al.* [18] have identified several representative behaviors of review spammers and have modeled these behaviors to detect the spammers. Mukherjee *et al.* [22] proposed a novel unsupervised model, named author spamicity model for discovering the opinion spammers by using their behavioral footprints. Also, Wu *et al.* [27] have studied the problem of detecting hybrid shilling attacks on rating data based on the semi-supervised learning algorithm. Moreover, Xie *et al.* [28] studied the problem of singleton review spam detection by detecting the co-anomaly patterns in multiple reviews based time series. While some of the previous studies used the popularity information in their applications, none of them could comprehensively model the popularity observations. Therefore, in this paper, we develop a novel approach for comprehensively modeling the popularity of mobile Apps. Particularly, the proposed PHMM is a general model and is applicable for most of the above applications.

Another category of related works is about the HMM based models, which have been widely used in various research domains, such as signal processing and speech recognition [5], [16], [24], biometrics [12], [17], [29], economics [4], [14], and search log mining [10]. For example, Rabiner [24] has proposed a comprehensive tutorial of HMM and its applications in speech recognition. Maiorana *et al.* [21] have applied the HMM into biometrics with the application of online signature recognition. Yamanishi and Maruyama [29] proposed to leverage HMM for network failure detection by estimating the anomaly sequences of system logs. Different from the above works, we study a new research problem, namely popularity modeling for mobile Apps, and propose a new model PHMM by extending the original HMM with heterogeneous popularity information.

III. Overview

Here, we first introduce the popularity observations of mobile Apps, and then present the problem statement of popularity modeling and the overview of our model.

A. Preliminaries of Popularity Observations

In this paper, we focus on three types of important popularity information, namely chart rankings, users ratings, and user reviews. We can collect periodical observations for each type of popularity information.

1) *Chart Rankings:* Most of the App stores provide the chart rankings of Apps, which are usually updated periodically, e.g., daily. Therefore, each mobile App a has many historical ranking observations which can be denoted as a time series, $\mathcal{P}_a = \{p_1^a, p_2^a, \ldots\}$, where $p_i^a \in \{1, \ldots, K_p\}$ is the ranking position of a at time stamp t_i. Note that, the smaller value p_i^a is, the higher ranking the App a has. Intuitively, the higher ranking indicates the higher popularity.
2) *User Ratings:* After an App is published, it can be rated by any user who has downloaded it. Indeed, the user rating is one of the most important features of App

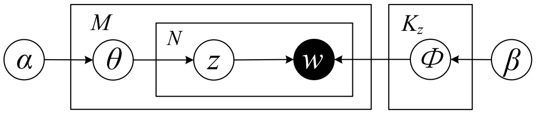


Fig. 1. Graphical representation of the LDA model.

popularity. Particularly, each rating can be categorized into K_r discrete rating levels, e.g., 1 to 5, which represent users' different preferences for Apps. Therefore, the rating observations of an App a can also be denoted as a time series, $\mathcal{R}_a = \{r_1^a, r_2^a, \ldots\}$, where $r_i^a \in \{1, \ldots, K_r\}$ is the rating posted at time stamp t_i.

3) *User Reviews:* In addition to ratings, users can also put their textual comments as reviews about each App. Each review reflects a user's personal perception for a particular App. Similar as user ratings, we can denote all reviews of an App a as a time series, $\mathcal{C}_a = \{c_1^a, c_2^a, \ldots\}$, where c_i^a is the review posted at time stamp t_i.

Indeed, all the above popularity observations are important for mobile App services. However, different from the ranking and rating observations, it is hard to directly leverage user reviews as observations for modeling App popularity. Therefore, in this paper, we propose to leverage topic modeling [8] to extract the latent semantics of user reviews as popularity observations [23], [31]. The intuitive motivation is mapping each review onto a specific review topic, which is easy to understand and exploit for opinion analysis. Specifically, in this paper, we adopt the widely used latent Dirichlet allocation (LDA) model [8] for learning latent semantic topics. To be more specific, the historical reviews of a mobile App a, i.e., $\mathcal{C}_a$, is assumed to be generated as follows. First, before generating $\mathcal{C}_a$, K_z prior conditional distributions of words given latent topics $\{\phi_z\}$ are generated from a prior Dirichlet distribution β. Second, a prior latent topic distribution θ_a is generated from a prior Dirichlet distribution α for each mobile App a. Then, for generating the jth word in $\mathcal{C}_a$ denoted as $w_{a,j}$, the model firstly generates a latent topic z from θ_a and then generates $w_{a,j}$ from ϕ_z. Particularly, Fig. 1 shows the graphical representation of the LDA model, where M is the number of mobile Apps, N is the number of all unique words in reviews, and K_z is the predefined number of latent topics. The training process of the LDA model is to learn proper latent variables $\theta = \{P(z|\mathcal{C}_a)\}$ and $\phi = \{P(w|z)\}$ for maximizing the posterior distribution of review observations, i.e., $P(\mathcal{C}_a|\alpha, \beta, \theta, \phi)$. In this paper, we use a Markov chain Monte Carlo method named Gibbs sampling [13] for training the LDA model. Then, we can map each review c_i^a onto a review topic z_i^a by

$$z_i^a = \arg\max_z P\left(z|c_i^a\right) \propto \arg\max_z \left(\prod_{w \in c_i^a} P\big(w|z\big)P(z)\right)$$

where both $P(w|z)$ and $P(z)$ can be learned via training the LDA model. Therefore, we can obtain the time series of reviews as $\mathcal{Z}_a = \{z_1^a, z_2^a, \ldots\}$, where $z_i^a \in \{1, \ldots, K_z\}$.

In reality, the time stamps of ranking observations and user rating (review) observations are usually not identical. For example, for a particular App, there may be multiple ratings and reviews from multiple users per day, while there may be only one ranking observation per day. Therefore, we first aggregate the ranking, and rating (review) observations together to get the integrated observations, which share the same time stamps. Moreover, in practice, we may need to model App popularity with different time granularity, e.g., daily or weekly. The aggregation allows us to get the integrated observations with different time interval/granularity. After getting the integrated observations with the same time stamps, we can represent the heterogeneous observations of popularity for each App a as a sequence $\mathcal{O}_a = \{o_1^a, o_2^a, \ldots\}$, where $o_i^a = \{\mathcal{P}_i^a, \mathcal{R}_i^a, \mathcal{Z}_i^a\}$ contains the observations of ranking, rating, and review topic during time interval t_i. Fig. 2 shows an example of extracting popularity observations, where each popularity observation contains different numbers of ranking, rating, and review topic.

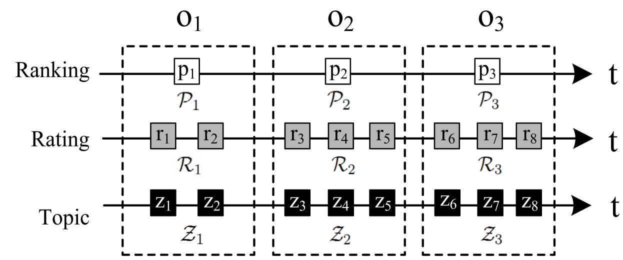


Fig. 2. Example of extracting popularity observations.

B. Problem Statement

Here, we formally define the problem of popularity modeling for mobile Apps.

Definition 1 (Problem Statement): Given a set of mobile Apps A, where each App $a \in A$ has a sequence of historical popularity observations $\mathcal{O}_a = \{o_1^a, o_2^a, \ldots\}$ ($|\mathcal{O}_a| \geq 1$). The problem of popularity modeling is to learn a model $\mathcal{M}$ from all the observation sequences $\{O_a|a \in A\}$, which can be used for predicting the popularity observations for each mobile App in the future.

However, it is not a trivial problem to model mobile App popularity. First, the popularity information of mobile Apps often varies frequently and has the instinct of sequence dependence. Second, the popularity information is heterogeneous but contains latent semantics and relationships. To solve these challenges, we propose a novel approach for popularity modeling based on HMM, which is widely used for modeling sequential observations. Specifically, we assume that there are multiple latent popularity states of mobile Apps, such as very popular, popular, and out-of-popular, and different kinds of popularity observations appear for one App at the same time because they all belong to the same popularity state with high probabilities (the probabilities will be learned by estimating the models). Moreover, the varying of popularity observations is due to the transitions of different popularity states. Fig. 3 shows the graphical representation of our PHMM. In this figure, each observation o_i is generated by the latent state s_i, and o_i contains $n_{p,i}$ rankings, $n_{r,i}$ ratings and $n_{z,i}$ review topics. Particularly, here, we adopt the widely used first order

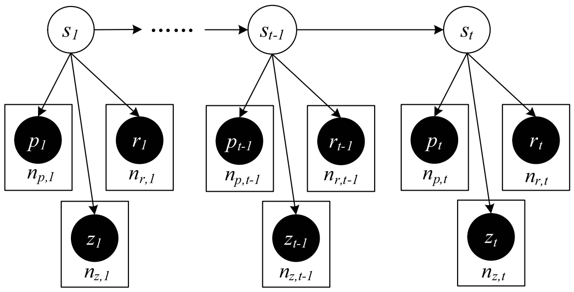


Fig. 3. Graphical representation of PHMM.

Markov assumption in our model. In other words, the probability distribution of each popularity state s_i is independent of the previous states $s_1, \ldots, s_{i-2}$, given the immediately previous state s_{i-1}, which means $P(s_i|s_1, \ldots, s_{i-1}) = P(s_i|s_{i-1})$.

In order to apply the proposed PHMM for modeling App popularity, there are two main challenges to be addressed. First, how to train PHMM from the historical heterogeneous popularity observations. Second, how to choose the proper number of latent popularity states for PHMM. In the following section, we will present our solutions for the above challenges.

IV. App Popularity Modeling

In this section, we present the details of App popularity modeling by the proposed PHMM.

A. Training of PHMM

Given a set of popularity states $\mathcal{S} = \{s_1, \ldots, s_{K_s}\}$, a set of App rankings $\mathcal{P} = \{p_1, \ldots, p_{K_p}\}$, a set of user ratings $\mathcal{R} = \{r_1, \ldots, r_{K_r}\}$, and a set of review topics $\mathcal{Z} = \{z_1, \ldots, z_{K_z}\}$, the proposed PHMM can be represented by a model containing three different probability distributions as follows.

1) The transition probability distribution $\Delta = \{P(s_i|s_j)\}$, where $s_i, s_j \in \mathcal{S}$ are the latent popularity states.
2) The initial state distribution $\Psi = \{P(s_i)\}$, where $P(s_i)$ is the probability that popularity state s_i occurs as the first element of a state sequence.
3) The emission probability distribution for each popularity state $\Lambda = \{P(\mathcal{P}, \mathcal{R}, \mathcal{Z}|s_i)\}$, where $P(\mathcal{P}, \mathcal{R}, \mathcal{Z}|s_i)$ is the joint probability that popularity observations $\mathcal{P}$, $\mathcal{R}$ and $\mathcal{Z}$ are generated by state s_i.

As the common setting of HMM, here we follow the "bag of words" assumption for PHMM, which assumes rankings, ratings, and review topics are conditionally independent given the popularity states [8], [10]. Formally, we have $P(\mathcal{P}, \mathcal{R}, \mathcal{Z}|s_i) \equiv \prod_{p\in\mathcal{P}} P(p|s_i) \prod_{r\in\mathcal{R}} P(r|s_i) \prod_{z\in\mathcal{Z}} P(z|s_i)$. Although, this assumption is relatively strong, it can effectively simplify the problem of popularity modeling and has some intuitive explanations. For example, given a mobile App, the appearance of high ratings may be not directly related to some specific review topics, but only depends on its current popularity state, e.g., "very popular."

Accordingly, we can further denote the emission probability distribution of different popularity observations, namely ranking, rating, and review topic, by the triple $(\Lambda_p, \Lambda_r, \Lambda_z) \equiv (\{P(p|s_i)\}, \{P(r|s_i)\}, \{P(z|s_i)\})$, which stratifies $\sum_p P(p|s_i) = \sum_r P(r|s_i) = \sum_z P(z|s_i) = 1$.

Therefore, given a set of training sequences of popularity observations $\mathcal{X} = \{\mathcal{O}_1, \ldots, \mathcal{O}_N\}$, the task of training PHMM is to learn the set of parameters $\Theta = (\Psi, \Delta, \Lambda_p, \Lambda_r, \Lambda_z)$. Specifically, we denote the length of sequence $\mathcal{O}_n$ as L_n and the jth observation $o_j \in \mathcal{O}_n$ as a triple $(\mathcal{P}_{n,j}, \mathcal{R}_{n,j}, \mathcal{Z}_{n,j})$. Moreover, we let $p_{n,j,k}$, $r_{n,j,k}$, and $z_{n,j,k}$ denote the kth ranking, rating, and review topic in $\mathcal{P}_{n,j}$, $\mathcal{R}_{n,j}$, and $\mathcal{Z}_{n,j}$. Therefore, we can use the maximum likelihood estimation (MLE) to compute the optimal parameters Θ^* by

$$\Theta^* = \arg\max_{\Theta} \log P(\mathcal{X}|\Theta) = \arg\max_{\Theta} \sum_n \log P(\mathcal{O}_n|\Theta). \quad (1)$$

Here, we denote $\mathcal{Y}_n \in \{S_1^n, \ldots, S_M^n\}$ as the state sequence of an observation sequence $\mathcal{O}_n$, where each S_m^n is a possible state sequence with length L_n. Moreover, we denote the jth state in S_m^n as $s_{m,j}^n$. Accordingly, we can define $\mathcal{Y} = \{\mathcal{Y}_1, \ldots, \mathcal{Y}_N\}$ as all latent variables of state sequences. Then, we can rewrite the likelihood $\log P(\mathcal{O}_n|\Theta)$ in (1) as $\log P(\mathcal{O}_n|\Theta) = \log \sum_m P(\mathcal{O}_n, S_m^n|\Theta)$, where the joint distribution can be written as shown at the bottom of this page.

Indeed, directly optimizing the above likelihood function is not a trivial problem. In this paper, we propose to exploit the expectation maximization (EM) algorithm to iteratively estimate the parameters.

Specifically, at the *E-step*, we have

$$\begin{aligned} Q\left(\Theta, \Theta^{(i-1)}\right) &= \mathbf{E}\left[\log P(\mathcal{X}, \mathcal{Y}|\Theta); \mathcal{X}, \Theta^{(i-1)}\right] \\ &= \sum_{n,m} P\left(S_m^n|\mathcal{O}_n, \Theta^{(i-1)}\right) \log P\left(\mathcal{O}_n, S_m^n|\Theta\right) \end{aligned} \quad (3)$$

where $\Theta^{(i-1)}$ is the set of model parameters estimated in the last round of EM iteration. Particularly, we can estimate $P(S_m^n|\mathcal{O}_n, \Theta^{(i-1)})$ by

$$P\left(S_m^n|\mathcal{O}_n, \Theta^{(i-1)}\right) = \frac{P\left(\mathcal{O}_n, S_m^n|\Theta^{(i-1)}\right)}{P\left(\mathcal{O}_n|\Theta^{(i-1)}\right)}. \quad (4)$$

$$\begin{aligned} P\left(\mathcal{O}_n, S_m^n|\Theta\right) &= P\left(\mathcal{O}_n|S_m^n, \Theta\right) P\left(S_m^n|\Theta\right) \\ &= \left(\prod_{j=1}^{L_n} \prod_k P\left(p_{n,j,k}|s_{m,j}^n\right) \prod_i P\left(r_{n,j,i}|s_{m,j}^n\right) \prod_t P\left(z_{n,j,t}|s_{m,j}^n\right)\right) \\ &\quad \times \left(P\left(s_{m,1}^n\right) \prod_{j=2}^{L_n} P\left(s_{m,j}^n|s_{m,j-1}^n\right)\right). \end{aligned} \quad (2)$$

At the *M-step*, we maximize function $Q(\Theta, \Theta^{(i-1)})$ iteratively until it converges by estimating the model parameters as follows:

$$P(s_i) = \frac{\sum_{m,n} P\left(S_m^n|\mathcal{O}_n, \Theta^{(i-1)}\right)\delta\left(s_{m,1}^n = s_i\right)}{\sum_{m,n} P\left(S_m^n|\mathcal{O}_n, \Theta^{(i-1)}\right)}$$

$$P(p|s_i) = \frac{\sum_{m,n} P\left(S_m^n|\mathcal{O}_n, \Theta^{(i-1)}\right)\sum_j \delta\left(s_{m,j}^n = s_i \wedge p \in \mathcal{P}_{n,j}\right)}{\sum_{m,n} P\left(S_m^n|\mathcal{O}_n, \Theta^{(i-1)}\right)\sum_j \delta\left(s_{m,j}^n = s_i\right)N_{\mathcal{P}_{n,j}}}$$

$$P(r|s_i) = \frac{\sum_{m,n} P\left(S_m^n|\mathcal{O}_n, \Theta^{(i-1)}\right)\sum_j \delta\left(s_{m,j}^n = s_i \wedge r \in \mathcal{R}_{n,j}\right)}{\sum_{m,n} P\left(S_m^n|\mathcal{O}_n, \Theta^{(i-1)}\right)\sum_j \delta\left(s_{m,j}^n = s_i\right)N_{\mathcal{R}_{n,j}}}$$

$$P(z|s_i) = \frac{\sum_{m,n} P\left(S_m^n|\mathcal{O}_n, \Theta^{(i-1)}\right)\sum_j \delta\left(s_{m,j}^n = s_i \wedge z \in \mathcal{Z}_{n,j}\right)}{\sum_{m,n} P\left(S_m^n|\mathcal{O}_n, \Theta^{(i-1)}\right)\sum_j \delta\left(s_{m,j}^n = s_i\right)N_{\mathcal{Z}_{n,j}}}$$

$$P(s_i|s_j) = \frac{\sum_{m,n} P\left(S_m^n|\mathcal{O}_n, \Theta^{(i-1)}\right)\delta\left(\exists t\ s_{m,t-1}^n = s_j \wedge s_{m,t}^n = s_i\right)}{\sum_{m,n} P\left(S_m^n|\mathcal{O}_n, \Theta^{(i-1)}\right)\delta\left(\exists t\ s_{m,t-1}^n = s_j\right)}$$

where $\delta(x) = 1$ if $x = True$, and 0 otherwise; $N_{\mathcal{P}_{n,j}}$, $N_{\mathcal{R}_{n,j}}$, and $N_{\mathcal{Z}_{n,j}}$ are the number of unique ranking, rating, topic observations in $\mathcal{P}_{n,j}$, $\mathcal{R}_{n,j}$, and $\mathcal{Z}_{n,j}$. Furthermore, the above equations can be efficiently computed by the forward–backward algorithm [24].

B. Choosing Number of Popularity States

Another problem of training PHMM is how to choose the proper number of latent popular states. Indeed, a common used approach for estimating the latent states of HMMs is to leverage domain knowledge or some existing algorithms to precluster the observations [10]. In our problem, the popularity observations contain $K_p + K_r + K_z$ elements, i.e., K_p unique rankings, K_r unique ratings, and K_z unique topics. Intuitively, these observations are heterogeneous and contain internal relationships. Thus, how to cluster such information is an open question. To solve this problem, in this paper, we propose a novel clustering method based on the element-record (E-R) bipartite graph. Specifically, the E-R bipartite graph can be denoted as $G = \{V, E, W\}$, where $V = \{V^b, V^o\}$. $V^b = \{b_1, \ldots, b_K\}$ denotes the set of unique observation elements, i.e., $K = K_p + K_r + K_z$, and $V^o = \{o_1, \ldots, o_M\}$ denotes the set of all observation records from sequence set $\mathcal{X}$. Edge set is $E = \{e_{ij}\}$, where e_{ij} denotes the observation element b_i has appeared in record o_j. Edge weight set is $W = \{w_{ij}\}$, where each w_{ij} represents the normalized frequency of the appearance of b_i in o_j. For example, if an observation element b_i = (Rating = 5) has appeared n_i times in o_j and there are totally n_j ratings in o_j, the weight would be $w_{ij} = n_i/n_j$. Fig. 4 shows an example of the E-R bipartite graph, which contains three popularity observations appeared in four records.

Therefore, given an E-R bipartite graph, we can denote each unique observation element as a normalized vector $\overrightarrow{b_i}$ = dim[M], where M is the number of all unique observation records, dim[j] $= w_{ij}/\sum_k w_{ik}$ is the normalized dimension of vector. Accordingly, we can estimate the similarity between two popularity observations by calculating the Cosine similarity between their vectors. After that, many existing algorithms can be leveraged for estimating the number of clusters, such as density based clustering algorithms [20]. In this paper, we utilize a clustering algorithm proposed in [11], which is robust for high dimensional data and only needs a parameter to indicate the minimum average mutual similarity $S_{\min}$ for the data points in each cluster. The average mutual similarity for a cluster C is calculated as

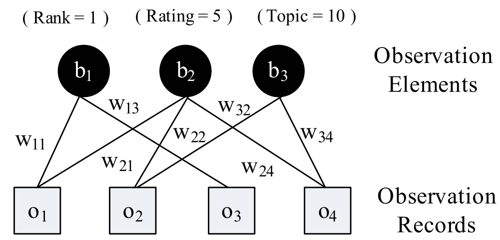


Fig. 4. Example of the E-R bipartite graph.

$$S_C = n\ \frac{2 \times \sum_{1 \le i < j \le |C|} \mathrm{Sim}\left(b_i, b_j\right)}{|C| \times \left(|C| - 1\right)} \tag{5}$$

where $|C|$ indicates the number of observation elements in C and Sim(b_i, b_j) is the Cosine similarity between the ith and jth observation elements in C.

The results of preclustering may not be the true popularity states learned by PHMM. However, we believe the preclustering can provide positive guidance for estimating the number of latent states due to the intrinsic relationships between popularity observations. Furthermore, the results of preclustering can also be used for assigning initial values of EM algorithm. Actually, the basic EM algorithm implemented by randomly assigning initial values for model parameters Θ, which may lead to more training iterations and unexpected local optimal results. Particularly, if we treat each popularity cluster C_i as the latent state s_i, we can estimate the initial values of parameters Θ as follows. First, we define the prior distribution of observation element b_i ($b = p, r, z$) as $P(b_i)$, which can be computed by the MLE method. Specifically, $P(b_i) = N_{b_i}/\sum_k N_{b_k}$, where N_{b_k} is the appearance frequency of b_k in all observation records. Second, for each observation element b_i, we can first compute the probability $P(s_j|b_i)$ by the normalized similarity between observation vector $\overrightarrow{b_i}$ and cluster C_j, i.e., $P(s_j|b_i) = (\mathrm{Sim}(\overrightarrow{b_i}, \overrightarrow{C_j}))/(\sum_k \mathrm{Sim}(\overrightarrow{b_i}, \overrightarrow{C_k}))$, where Sim$(*, *)$ is the Cosine similarity and $\overrightarrow{C_j} = (\sum_{b \in C_i} \overrightarrow{b})/|C_i|$ is the centroid of cluster C_j. Actually, here we assume that an observation has higher probability of belonging to a nearer cluster. Therefore, we have following estimations.

1) The initial state distribution can be computed by $P^0(s_i) = \sum_{b=p,r,z}\sum_k P(s_i|b_k)P(b_k)$.
2) The emission probability can be computed according to the Bayes rule, i.e., $P^0(b_i|s_j) = (P(s_j|b_i)P(b_i))/(P^0(s_j))$.
3) The transition distribution $P^0(s_i|s_j)$ can be computed by the normalized similarity between cluster C_i and C_j, i.e., $P^0(s_i|s_j) = (\mathrm{Sim}(\overrightarrow{C_i}, \overrightarrow{C_j}))/(\sum_k \mathrm{Sim}(\overrightarrow{C_k}, \overrightarrow{C_j}))$.

Indeed, our experimental results clearly validate that using preclustering for assigning initial values of EM algorithm can accelerate the training process and enhance the model fitting of PHMM.

V. Applications of PHMM

Indeed, many applications can be derived from the proposed PHMM. In the following, we demonstrate three motivating examples including trend based mobile App recommendation, rating and review spam detection, as well as ranking fraud detection for mobile Apps.

Particularly, PHMM can be learned from different model granularity. For example, as introduced in Section III, we can use the daily, weekly or monthly observations from one or more Apps for modeling popularity. Different model granularity may generate different popularity patterns and lead to different applications. In this paper, we mainly focus on learning PHMM from all mobile Apps, which can be used for capturing the common popularity patterns and relationships of Apps.

Assume that we have observed a sequence of popularity observations $\mathcal{O}_a = \{o_1^a, \ldots, o_t^a\}$ from a mobile App a. Thus, we can estimate the latent states for the ith observation o_i^a $(0 \leq i \leq t)$ by

$$\begin{aligned} P\left(s|o_i^a, \Theta\right) &\propto P\left(o_i^a|s, \Theta\right) P\left(s|\Theta\right) \\ &= \prod_{p,r,z \in o_i^a} P\left(p|s\right) P\left(r|s\right) P\left(z|s\right) \sum_{s'} P\left(s|s'\right) P\left(s'|o_{i-1}^a, \Theta\right) \end{aligned} \quad (6)$$

which can be computed effectively by Forward–Backward algorithm or Viterbi algorithm [26]. Similarly, we can predict the $(t+1)$-st popularity state for App a by computing

$$P\left(s^{(t+1)} = s|\mathcal{O}_a, \Theta\right) = \sum_{s'} P\left(s|s'\right) P\left(s'|o_t^a, \Theta\right). \quad (7)$$

Based on the above, we can conduct the following three mobile App related applications.

A. Trend-Based Mobile App Recommendation

The existing mobile App recommender systems usually recommend Apps, which were popular in the past. This is not proper in practice, because the popularity information is always varying frequently, and mobile users tend to follow the future popularity trend of Apps. Therefore, in this paper, we propose a trend based App recommendation approach by leveraging PHMM. Specifically, given a t-length observation sequence of mobile App a, i.e., $\mathcal{O}_a = \{o_1^a, \ldots, o_t^a\}$, we can predict the possible rankings and ratings of a at next time stamp $t+1$ by $P(p^{(t+1)} = p|\mathcal{O}_a, \Theta) = \sum_s P(p|s)P(s^{(t+1)} = s|\mathcal{O}_a, \Theta)$, and $P(r^{(t+1)} = r|\mathcal{O}_a, \Theta) = \sum_s P(r|s)P(s^{(t+1)} = s|\mathcal{O}_a, \Theta)$, where $s^{(t+1)}$ is the $(t+1)$-st popularity state of $\mathcal{O}_a$. Furthermore, we can compute the ranking and rating expectations of App a at time stamp $t+1$ by $p_a^* = \sum_p p \times P(p^{(t+1)} = p|\mathcal{O}_a, \Theta)$ and $r_a^* = \sum_r r \times P(r^{(t+1)} = r|\mathcal{O}_a, \Theta)$. Similarly, we can rank all mobile Apps with respect to their ranking and rating expectations, and obtain two ranked list Υ_{Rank} and Υ_{Rate}. Then, we can calculate the final popularity score of each mobile App by Borda's ranking fusion method

$$P\_Score(a) = \alpha \times \frac{1}{RK_{\text{Rank}}(a)} + (1-\alpha) \times \frac{1}{RK_{\text{Rate}}(a)} \quad (8)$$

where α is the fusion parameter; $RK_{\text{Rank}}(a)$ and $RK_{\text{Rate}}(a)$ is the ranking of a in ranked list Υ_{Rank} and Υ_{Rate}. Particularly, if $\alpha = 0$, the final rank is only based on the rating trend, which is similar to the ranked list Υ_{Rate}. If $\alpha = 1$, the final rank is only based on the ranking trend, which is similar to the ranked list Υ_{Rank}. The score $P\_Score(a)$ indicates the popularity trend in the future, thus can be used for recommending Apps.

B. Rating and Review Spam Detection

User ratings and reviews are the important information in mobile App market. The App store provider and the developers of Apps rely on the ratings and reviews a lot to get helpful feedback from various users. However, some of the shady users may post deceptive ratings and reviews with the purpose of inflating or deflating corresponding mobile Apps. Many efforts have been made in the literatures for detecting such rating and review spams [18], [27], [28]. However, few of them took the sequence characteristics of App popularity into consideration. In this paper, we propose a novel approach based on PHMM for detecting rating and review spams. Specifically, given a t-length observation sequence of mobile App a, i.e., $\mathcal{O}_a = \{o_1^a, \ldots, o_t^a\}$, we can first leverage the first $(t-1)$-length sequence to predict the possible tth popularity states of a by (6). Then, we can calculate the likelihood of the observations of rating and review topic at time stamp t by $\log P(\mathcal{R}_t^a|\Theta) = \log \sum_s P(\mathcal{R}_t^a, s^t = s|\Theta)$, and $\log P(\mathcal{Z}_t^a|\Theta) = \log \sum_s P(\mathcal{Z}_t^a, s^t = s|\Theta)$, which can be estimated by the similar way of (2). Then, if the likelihood is less than the predefined thresholds τ_r, and τ_z, we believe that there are rating or review spams in a at time stamp t.

C. Ranking Fraud Detection

The ranking fraud of mobile Apps refers to fraudulent or deceptive activities, which have a purpose of bumping up the rankings of Apps during a specific time period [33]. Detecting such ranking fraud is very important for the healthy development of mobile App industry, especially for building mobile App recommender systems. Different from rating and review spam, the ranking fraud always happens during some specific time periods. It is due to that people, who try to manipulate the App rankings always have some specific expectations of ranking, such as top 25 for one month. Moreover, some of the normal promotion means, such as version update, may also result in the abnormal ranking observations. Therefore, to accurately detect the ranking fraud for mobile Apps, we should check the observation sequence during a time period but not at only one time stamp. To be specific, we can first define a sliding window with length T, and segment the popularity records of mobile Apps a by several T-length observation sequences $\{\mathcal{O}_1^a, \ldots, \mathcal{O}_n^a\}$. And then, for each sequence $\mathcal{O}_i^a$ we

TABLE I
SOME STATISTICS OF THE EXPERIMENTAL DATA

	Top-Free 300	Top-Paid 300
App Num.	9,784	5,261
Ranking Num.	285,900	285,900
Avg. Ranking Num.	29.22	54.34
Rating/Review Num.	14,912,459	4,561,943
Avg. Rating/Review Num.	1,524.17	867.12

will calculate its anomaly score by the average log-loss of ranking observations

$$\mathcal{L}\left(\mathcal{O}_i^a\right) = -\frac{1}{T}\log P\left(\mathcal{P}_{\mathcal{O}_i^a}|\Theta\right) = -\frac{1}{T}\log\sum_m P\left(\mathcal{P}_{\mathcal{O}_i^a}, S_m^T|\Theta\right) \tag{9}$$

where $\mathcal{P}_{\mathcal{O}_i^a} = \{\mathcal{P}_{i,1}^a, \ldots, \mathcal{P}_{i,T}^a\}$ is the sequence of all ranking observations in $\mathcal{O}_i^a$, and each S_m^T is a state sequence with length T and the equation can be estimated in the similar way as (2). Finally, if the anomaly score $\mathcal{L}(\mathcal{O}_i^a)$ is larger than a predefined threshold τ_p, we believe that the ranking fraud happens during the time period of $\mathcal{O}_i^a$.

VI. EXPERIMENTAL RESULTS

In this section, we evaluate the performance of PHMM by using two real-world App data sets.

A. Experimental Data

The experimental data sets were collected from the "Top Free 300" and "Top Paid 300" leaderboards of Apple's Apps Store (U.S.) from February 2, 2010 to September 17, 2012. The data set contains the daily chart rankings, which were collected at 11:00 P.M. (PST) daily, all user ratings and user reviews of the top-300 free Apps and the top-300 paid Apps during the period, respectively. Moreover, we also know the initial release date of each App, which can be leveraged for determining the first observation record. To be specific, Table I shows the detailed statistics of our two data sets. Note that, since each rating record is always associated with a user review in Appstore, the number of reviews is always the same as that of ratings for Apps in our data sets.

Fig. 5(a)–(c) shows the distribution of the number of Apps with respect to different rankings, different rating levels and different number of ratings/reviews in the Top-Free 300 data set. From these figures we can find that the distributions of popularity observations are not even. For example, most of the Apps have average ratings higher than 3. Furthermore, we used the LDA model to extract review topics as introduced in Section II. Particularly, we first removed all stop words and normalized verbs and adjectives of each review by the stop-words remover and the Porter stemmer [31]. Then, the number of latent topic K_z was set to 20 according to the perplexity based estimation approach [6], [7]. Two parameters α and β for training LDA model were set to be $50/K$ and 0.1 according to [15]. Fig. 5(d) shows the distribution of the number of reviews with respect to different topics. From this figure we can observe that only a few topics are frequently mentioned

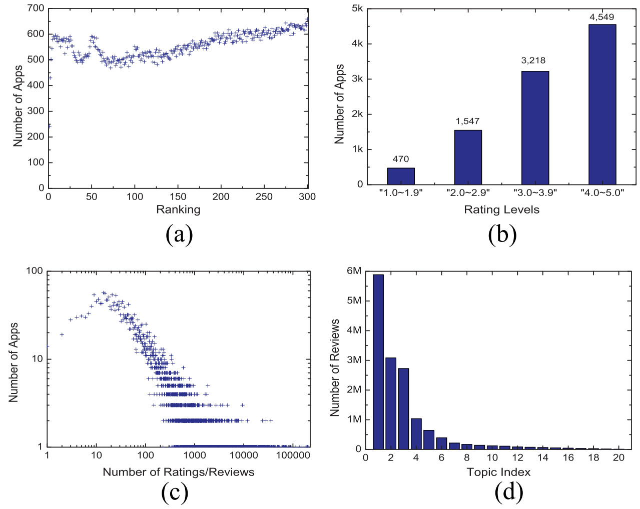


Fig. 5. Data distributions in the (a)–(d) Top-Free 300 data set.

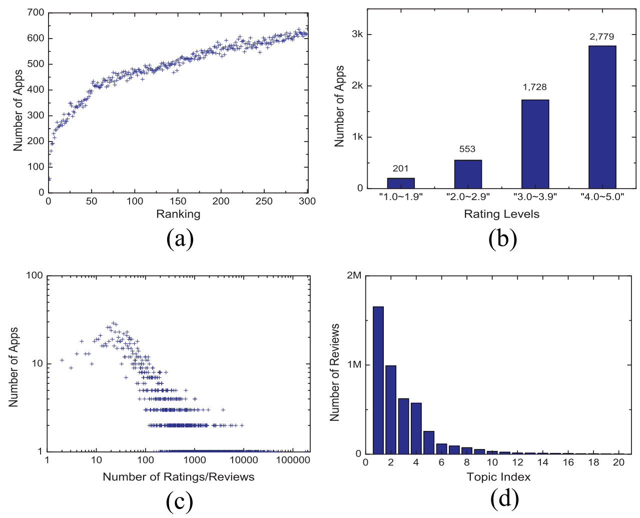


Fig. 6. Data distributions in the (a)–(d) Top-Paid 300 data set.

in reviews. In addition, Fig. 6 shows the similar results of the Top-Paid 300 data set.

B. Performance of Training PHMM

In this subsection, we demonstrate the performance of training PHMM. In particular, we treat the data of each day as an observation record. Therefore, for each observation record of a specific App, there are one chart ranking, several user ratings and user reviews (topics).

We first used the approach introduced in Section IV-B for preclustering popularity observations from all Apps, where the parameter $S_{\min}$ is empirically set as 0.5. After that, there are totally 13 and 12 clusters used for assigning initial values of PHMM parameters in the Top-Free 300 and Top-Paid 300 data sets, respectively. Fig. 7 shows the value of the $Q(\Theta, \Theta^{(i-1)})$ of PHMM with initial value assignment of model parameters randomly, and initial value assignment of model parameters by preclustering in each iteration with respect to two data sets. Particularly, both experiments are conducted five times and the figures show the average values. From these figures, we can observe that the preclustering approach can generate

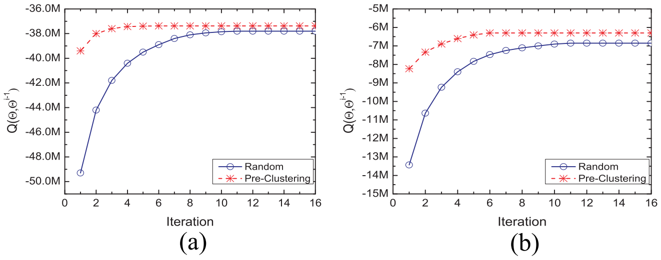


Fig. 7. Value of the $Q(\Theta, \Theta^{(i-1)})$ of PHMM with different initial value assignment in two data sets. (a) Top-Free 300. (b) Top-Paid 300.

TABLE II
STATE s_1 IDENTIFIED IN THE TOP-FREE 300 DATA SET

Ranking=5
Ranking=15
Rating=5
Rating=4
Topic="Funny Apps"
Topic="Good Design"

TABLE III
STATE s_6 IDENTIFIED IN THE TOP-FREE 300 DATA SET

Ranking=253
Ranking=148
Rating=3
Rating=4
Topic="Boring"
Topic="Old-Fashioned"

higher initial value of the $Q(\Theta, \Theta^{(i-1)})$, and thus dramatically accelerate the training process of PHMM. Moreover, we also observe that PHMM with initial value assignment by the preclustering can achieve the better model fitting, which indicates the higher likelihood of PHMM.

To further demonstrate the performance of PHMM, we also show four examples of different identified latent popularity states from two data sets in Tables II–V. Note that, here we manually transferred each review topic to semantic description, and due to the limited space we only show top 2 ranking, rating, and topic observations which are most probable to appear in each state. From the tables, we can observe that the latent popularity states are meaningful. For example, the states s_1 and s_6 identified by PHMM in the Top-Free 300 data set may indicate the Apps are in the states very popular and out-of-popular, respectively.

In addition, it is interesting to analyze the transition probability of popularity states, i.e., which kinds of popularity states are more easily to transit to others? Indeed, this question can be generalized as the estimation of the dwell time of each popularity state. The states having a short dwell time will transit to other states easily, and the popularity trend will stay for a while at the states having long dwell time. Specifically, it is easy to prove that the dwell time ω_s of the popularity state s is proportional to the transition probability that state s to itself, i.e., $P(s|s)$. By inspecting the training results of PHMM, we observe that the dwell times of the states that have high probability of emitting high ranking and ratings (e.g., the state shown in Table II) are usually very short, and vice versa. It

TABLE IV
STATE s_2 IDENTIFIED IN THE TOP-PAID 300 DATA SET

Ranking=50
Ranking=45
Rating=4
Rating=5
Topic="Recommended"
Topic="Funny Apps"

TABLE V
STATE s_8 IDENTIFIED IN THE TOP-PAID 300 DATA SET

Ranking=115
Ranking=153
Rating=4
Rating=3
Topic="Recommended"
Topic="Improvement Needed"

might be due to the fierce competition in the App market, and most Apps only could be popular for a short period. Particularly, with the estimation of dwell time, developers can obtain potential insights on how to promote their Apps. For example, if an App is identified in a very popular latent state, whose dwell time is short, the developer can design some marketing campaigns in advance for keeping the momentum.

C. Effectiveness of PHMM

Indeed, all the potential applications of PHMM introduced in Section IV are based on the prediction of popularity observations. Therefore, in this subsection, we validate the effectiveness of PHMM by evaluating its performance of predicting rankings, rankings and topics. To reduce the uncertainty of splitting the data into training and test data, in the experiments we utilized the fivefold cross validation to evaluate PHMM, which is widely used for various information retrieval tasks. To be specific, we first randomly divided all observation sequences of mobile Apps into five equal parts, and then used each part as the test data while using other four parts as the training data in five test rounds. Finally, we evaluated the results with different metrics in each test round. Particularly, for each test sequence, we randomly selected top T observation records for fitting model, and used the $(T+1)$-st observation record as ground truth for predicting probable rankings, ratings, and topics. Moreover, in our experiments, we found that the preclustering results of each training data set are very similar, thus, we set the number of popularity states as 13 for all five PHMMs in the Top-Free 300 data set and 12 for all five PHMMs in the Top-Paid 300 data set, respectively. Indeed, the cross validation might neglect the effect of temporality of model training. However, due to the limitation of our data collection, the data distributions in two data sets are unbalanced, it is hard to find proper time stamps for accurately segmenting training and test data. Therefore, here we use the widely used cross validation for evaluating the overall performance of popularity prediction.

To our best knowledge, there is no existing work of App popularity modeling has been reported. Therefore, here we developed two advanced baselines for evaluating our PHMM, which are static and sequential approaches, respectively.

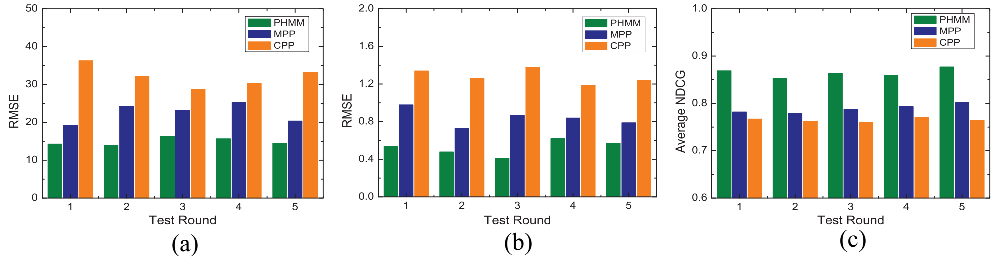


Fig. 8. Performances of predicting popularity observations by each approach in the Top-Free 300 data set. (a) Ranking prediction. (b) Rating prediction. (c) Topic prediction.

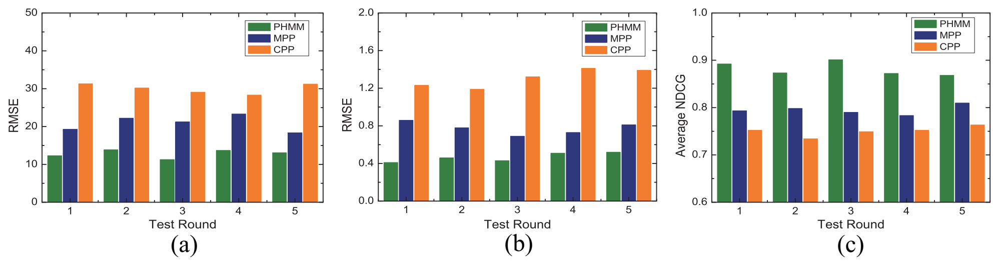


Fig. 9. Performances of predicting popularity observations by each approach in the Top-Paid 300 data set. (a) Ranking prediction. (b) Rating prediction. (c) Topic prediction.

The first baseline CPP stands for preclustering based popularity prediction, which is a static approach for predicting popularity observations. To be specific, given a T-length observation sequence $\mathcal{O} = \{o_1, \ldots, o_T\}$, we predict the popularity observation $b^{(T+1)}$ $(b = p, r, z)$ by the probability $P(b^{(T+1)} = b|\mathcal{O}) = \sum_i P(b^{(T+1)} = b, C_i|\mathcal{O}) \propto \sum_i P(b^{(T+1)} = b|C_i)P(C_i|\mathcal{O})$, where C_i is the ith observation cluster. Meanwhile, $P(b^{(T+1)} = b|C_i)$ can be estimated by the approach introduced in Section III, and $P(C_i|\mathcal{O})$ can be computed by $P(C_i|\mathcal{O}) \propto P(C_i) \prod_{j=1}^{T} \prod_{k,m,n} P(p_{j,k}|C_i)P(r_{j,m}|C_i)P(z_{j,n}|C_i)$, where $p_{j,k}$, $r_{j,m}$, and $z_{j,n}$ denote the kth ranking observation, mth rating and nth topic observations in observation record $o_j \in \mathcal{O}$, respectively.

The second baseline MPP stands for Markov chain based popularity prediction, which is a sequential approach with first order Markov assumption. Specifically, given a T-length observation sequence $\mathcal{O} = \{o_1, \ldots, o_T\}$, we predict the popularity observation $b^{(T+1)}$ $(b = p, r, z)$ by the probability $P(b^{(T+1)} = b|\mathcal{O}) = P(b^{(T+1)} = b|o_T)$. We have the probability $P(b^{(T+1)} = b|o_T) \propto P(b^{(T+1)} = b) \prod_{k,m,n} P(p_{T,k}|b^{(T+1)} = b)P(r_{T,m}|b^{(T+1)} = b)P(z_{T,n}|b^{(T+1)} = b)$, where probabilities $P(b^{(T+1)} = b)$ and $P(b'|b^{(T+1)} = b)$ $(b' = p, r, z)$ can be computed by the MLE method. Specifically, we have $P(b^{(T+1)} = b) = N_b / \sum_{b'} N_{b'}$, and $P(b'|b^{(T+1)} = b) = N^o_{b',b}/N^o_b$, where N_b is the frequency of b in all observation records. Meanwhile, N^o_b is the number of observation records in training data that contain observation b, and $N^o_{b',b}$ is the number of observation records in training data that contain b, where the last records contain observation b'.

First, we compared the performance of ranking and rating prediction by each approach. Indeed, both ranking and rating are numerical observations, thus, we expect the prediction results should be close to the ground truth values of observations. Particularly, in our data sets, there are one ranking observation and several rating observations in each observation record. Therefore, we can use the ranking observation and the average rating as ground truth values for evaluation. Specifically, we evaluate each approach by calculating the root mean square error (RMSE) with the predicted results and ground truth values for all test sequences. Take ranking as an example, we define RMSE $= \sqrt{(\sum_{\mathcal{O}_i} (p_i^* - b_i^{\Delta})^2)/N}$, where $\mathcal{O}_i$ is the ith test sequence with length T_i, $p_i^* = \arg\max_p P(p^{(T_i+1)} = p|\mathcal{O}_i)$, p_i^{Δ} is the ground truth ranking in the $(T_i + 1)$-st observation record, and N is the number of test sequences. Moreover, we can calculate the RMSE of rating prediction in a similar way. The smaller RMSE value, the better performance of ranking and rating prediction. Figs. 8(a) and (b) and 9(a) and (b) show the RMSE results of the ranking and rating prediction of each approach in the five test rounds of the two data sets. From this figure, we can observe that the PHMM has the best RMSE performance, which means it can predict the most accurate rankings and ratings for mobile Apps. Moreover, the sequential approach MPP outperforms static approach CPP, which indicates that the sequence dependence is an important characteristic of popularity observation.

Second, we evaluated the performance of predicting topics by each approach. Different from ranking and rating, review topic is categorical observation. Therefore, we propose to exploit the popular metric normalized discounted cumulative gain (NDCG) for evaluation. Specifically, in the ground truth observation records, there are several review topics, thus we define the ground truth relevance of each unique topic z, i.e., $Rel(z)$, as its normalized appearance frequency in the record. Also, each approach can predicate a ranked list, i.e., Υ_{PR}, of topics for each test sequence $\mathcal{O}_i$ with respect to the posterior probability $P(z^{(T_i+1)} = z|\mathcal{O}_i)$. After that, we can calculate the discounted cumulative gain (DCG) of each approach by DCG $= \sum_{i=1}^{K_z} (2^{Rel(z_i)} - 1)/(\log_2(1 + i))$, where $K_z = 20$ is the number of topics, z_i is the ith topic in Υ_{PR}, $Rel(z_i)$ is the

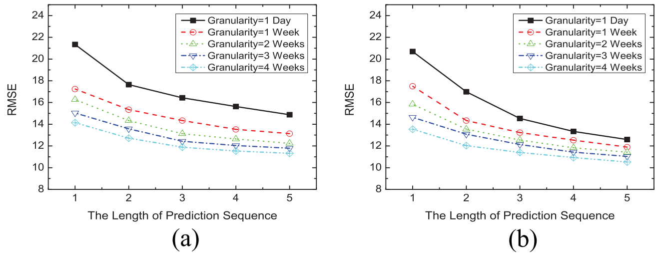


Fig. 10. Performance of the ranking prediction of PHMM with respect to different model granularity and sequence length in two data sets. (a) Top-Free 300. (b) Top-Paid 300.

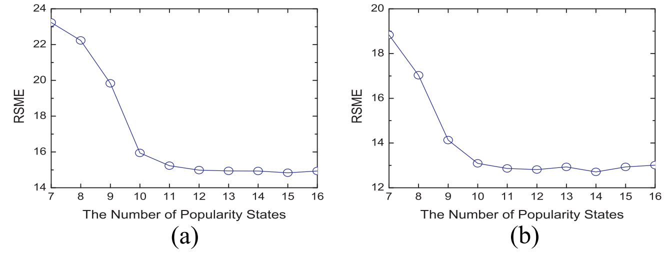


Fig. 11. Performance of the ranking prediction of PHMM with respect to different numbers of popularity states in two data sets. (a) Top-Free 300. (b) Top-Paid 300.

ground truth relevance. The NDCG is the DCG normalized by the IDCG, which is the DCG value of the ideal ranking list of the returned results and NDCG = DCG/IDCG.

Finally, we calculate the average NDCG for all test cases. Indeed, NDCG indicates how well the rank order of topics is by each approach. The larger NDCG value, the better performance of topic prediction. Figs. 8(c) and 9(c) show the NDCG results of each approach in the five test rounds of the two data sets. We can observe that the result trend is similar as that of ranking and rating prediction, where our PHMM have the best prediction performance and sequential approach MPP outperforms CPP.

Based on the above experimental results, we can have two conclusions as follows. First, our PHMM is an effective approach to model popularity observations for mobile Apps, since it can capture the semantic and sequence characteristics of the popularity observations of mobile Apps. Second, the sequence characteristic is important for App popularity modeling, thus we should take consideration of this for related services.

D. Robustness of PHMM

In this subsection, we will further study some important model parameters, which might affect the prediction performance of PHMM.

1) Effects of Granularity and Sequence Length: As mentioned in Section III, PHMM allows to aggregate popularity observations with different granularity (e.g., daily and weekly) for satisfying various needs of popularity modeling. Meanwhile, it is intuitive that different model granularity will result in popularity sequences with different length, which may affect the prediction performance of PHMM. Therefore, it is important to evaluate the effect of granularity and sequence length on popularity prediction of mobile Apps. Specifically, Fig. 10 shows the performance of the ranking prediction of PHMM with respect to different model granularity and length of prediction sequence in two data sets. Note that, here we only demonstrate the results of ranking prediction because the results of rating/review prediction have the similar statistical trends. All the results are the average RMSE values obtained by the fivefold cross validation. From this figure, we can find that as the granularity and the length of prediction sequence increase, the RMSE values decrease and almost reach the optimal values after the sequence length is larger than 4. It might indicate that longer prediction sequence and larger model granularity contain more information on the popularity dynamics, and can capture the sequence dependence of popularity more accurately. Therefore, choosing relatively large model granularity and long sequence length can improve the prediction performance of PHMM.

2) Effect of State Number: Indeed, the proposed PHMM needs a parameter to determine the number of popularity states. Although our bipartite graph based clustering approach can be leveraged for estimating this parameter, they are still not clear whether this approach is robust enough and how is the impact of the parameter on PHMM. To this end, we also tested the robustness of PHMM with varying settings of the number of popularity states. Fig. 11(a) and (b) shows the performance of the ranking prediction (i.e., average RMSE) of PHMM with respect to varying number of popularity states in two data sets. From these figures we can observe that the performance of ranking prediction is not good with small numbers of popularity states (i.e., RMSE decreases dramatically with the increase of parameter), while it becomes stable when the setting of the number increases. The phenomenon is reasonable, since small number of popularity states cannot fully capture the semantics of popularity observations. The results also validates the effectiveness of our parameter estimation approach.

E. Case Study of Ranking Fraud Detection

As introduced in Section IV, PHMM can be used for detecting ranking fraud for mobile Apps. Here, we study the performance of ranking fraud detection based on the prior knowledge from existing reports. Specifically, as reported by IBTimes [3], there are eight free Apps which might involve the ranking fraud. In this paper, we used seven of them in our data set [33] for evaluation. Particularly, instead of using sliding window to segment observation sequences, we directly calculated the anomaly score with respect to all popularity observations of each sequence. When we ranked all Apps in our data set with respect their ranking anomaly scores, we found that all above suspicious Apps are ranked in top 5%, which indicates PHMM can find these suspicious Apps with high rankings. Moreover, Fig. 12(a) and (b) shows the ranking records of the highest-ranked (i.e., most suspicious) Apps in our data set and one of the seven suspicious Apps. From these figures, we can find that the Apps contain several impulsive ranking patterns with high ranking positions. In contrast, the ranking behaviors of the normal Apps may be completely different. For example, Fig. 12(c) and (d) shows the ranking records of the lowest-ranked (i.e., most normal) App in our data set and a popular App "Angry Birds: Season-Free," both of which have the clear popularity trends. In fact, once a normal App is ranked high

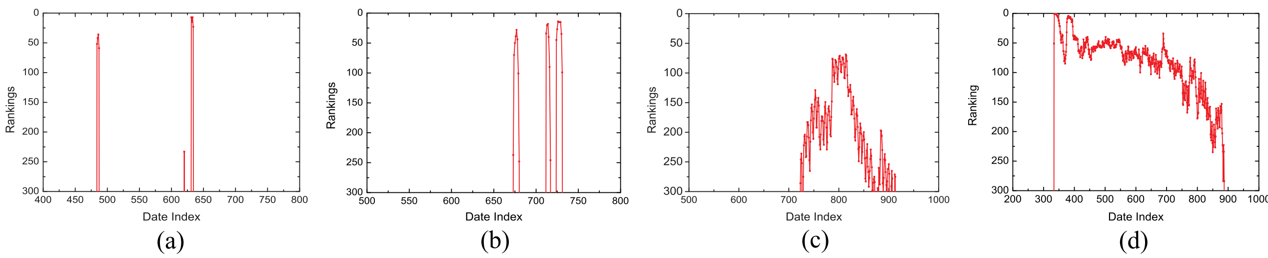


Fig. 12. Demonstration of the ranking records of four different mobile Apps. (a) App 1. (b) App 2. (c) App 3. (d) App 4.

in the leaderboard, it often owns a lot of honest fans and may attract more and more users to download. Thus, the popularity information will not vary dramatically in a short time.

VII. Conclusion

In this paper, we presented a sequential approach for modeling the popularity information of mobile Apps. Along this line, we first proposed a PHMM by learning the sequences of heterogeneous popularity observations from mobile Apps. Then, we introduced a bipartite based method to precluster the popularity observations. This can efficiently learn the parameters and initial values of PHMM. A unique perspective of our approach is that it can capture the sequence dependence and the latent semantics of multiple popularity observations. Furthermore, we also demonstrated some mobile App services enabled by PHMM, including trend based App recommendation, rating and review spam detection, and ranking fraud detection. Finally, the extensive experiments on two real-world data sets collected from the Apple Appstore clearly showed the efficiency and effectiveness of our approach.

Indeed, PHMM is a general model for modeling multiple sequential observations and can be easily extended to other related domains, such as product modeling in E-commerce. Thus, in the future, we plan to study more applications of PHMM in these domains. Moreover, we would like to explore more potential popularity factors to improve the modeling performance of PHMM.

References

[1] (2014). *Google Play* [Online]. Available: http://play.google.com

[2] (2014). *Apple Appstore* [Online]. Available: http://www.apple.com/iphone/from-the-app-store/

[3] (2012). *IBTimes* [Online]. Available: http://www.ibtimes.com/apple-threatens-crackdown-biggest-app-store-ranking-fraud-406764

[4] J. H. Albert and S. Chib, "Bayes inference via Gibbs sampling of autoregressive time series subject to Markov mean and variance shifts," *J. Bus. Econ. Stat.*, vol. 11, no. 1, pp. 1–15, 1993.

[5] C. Andrieu and A. Doucet, "Joint Bayesian model selection and estimation of noisy sinusoids via reversible jump MCMC," *IEEE Trans. Signal Process.*, vol. 47, no. 10, pp. 2667–2676, Oct. 1999.

[6] L. Azzopardi, M. Girolami, and K. V. Risjbergen, "Investigating the relationship between language model perplexity and IR precision-recall measures," in *Proc. SIGIR*, Toronto, ON, Canada, 2003, pp. 369–370.

[7] T. Bao, H. Cao, E. Chen, J. Tian, and H. Xiong, "An unsupervised approach to modeling personalized contexts of mobile users," in *Proc. IEEE 10th Int. Conf. Data Mining (ICDM)*, Sydney, NSW, Australia, 2010, pp. 38–47.

[8] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Lantent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Jan. 2003.

[9] M. Böhmer, L. Ganev, and A. Krüger, "AppFunnel: A framework for usage-centric evaluation of recommender systems that suggest mobile applications," in *Proc. Int. Conf. Intell. User Interfaces (IUI)*, New York, NY, USA, 2013, pp. 267–276.

[10] H. Cao, D. Jiang, J. Pei, E. Chen, and H. Li, "Towards context-aware search by learning a very large variable length hidden Markov model from search logs," in *Proc. 18th Int. Conf. World Wide Web (WWW)*, New York, NY, USA, 2009, pp. 191–200.

[11] H. Cao *et al.*, "Context-aware query suggestion by mining click-through and session data," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining (KDD)*, New York, NY, USA, 2008, pp. 875–883.

[12] D. R. Fredkin and J. A. Rice, "Bayesian restoration of single-channel patch clamp recordings," *Biometrics*, vol. 48, pp. 427–448, Jun. 1992.

[13] T. L. Griffiths and M. Steyvers, "Finding scientific topics," in *Proc. Nat. Acad. Sci. USA*, 2004, pp. 5228–5235.

[14] J. D. Hamilton, "A new approach to the economic analysis of non-stationary time series and the business cycle," *Econometrica*, vol. 57, pp. 357–384, Mar. 1989.

[15] G. Heinrich. (2008). "Parameter estimation for text analysis," Dept. Comput. Sci., University of Leipzig, Leipzig, Germany, Tech. Rep. [Online]. Available: http://faculty.cs.byu.edu/~ringger/CS601R/papers/Heinrich-GibbsLDA.pdf

[16] B.-H. Juang and L. R. Rabiner, "Hidden Markov models for speech recognition," *Technometrics*, vol. 33, no. 3, pp. 251–272, 1991.

[17] B. G. Leroux and M. L. Puterman, "Maximum-penalized-likelihood estimation for independent and Markov-dependent mixture models," *Biometrics*, vol. 48, pp. 545–558, Jun. 1992.

[18] E.-P. Lim, V.-A. Nguyen, N. Jindal, B. Liu, and H. W. Lauw, "Detecting product review spammers using rating behaviors," in *Proc. 19th ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, New York, NY, USA, 2010, pp. 939–948.

[19] Q. Liu, H. Ma, E. Chen, and H. Xiong, "A survey of context-aware mobile recommendations," *Int. J. Inf. Technol. Decis. Making*, vol. 12, no. 1, pp. 139–172, 2013.

[20] Y. Liu *et al.*, "Understanding and enhancement of internal clustering validation measures," *IEEE Trans. Cybern.*, vol. 43, no. 3, pp. 982–994, Jun. 2013.

[21] E. Maiorana, P. Campisi, J. Fierrez, J. Ortega-Garcia, and A. Neri, "Cancelable templates for sequence-based biometrics with application to on-line signature recognition," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 40, no. 3, pp. 525–538, May 2010.

[22] A. Mukherjee *et al.*, "Spotting opinion spammers using behavioral footprints," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining (KDD)*, New York, NY, USA, 2013, pp. 632–640.

[23] X.-H. Phan *et al.*, "A hidden topic-based framework toward building applications with short web documents," *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 7, pp. 961–976, Jul. 2010.

[24] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.

[25] K. Shi and K. Ali, "Getjar mobile application recommendations with very sparse datasets," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining (KDD)*, New York, NY, USA, 2012, pp. 204–212.

[26] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. Inf. Theory*, vol. 13, no. 2, pp. 260–269, Apr. 1967.

[27] Z. Wu, J. Wu, J. Cao, and D. Tao, "HySAD: A semi-supervised hybrid shilling attack detector for trustworthy product recommendation," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining (KDD)*, New York, NY, USA, 2012, pp. 985–993.

[28] S. Xie, G. Wang, S. Lin, and P. S. Yu, "Review spam detection via temporal pattern discovery," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining (KDD)*, New York, NY, USA, 2012, pp. 823–831.

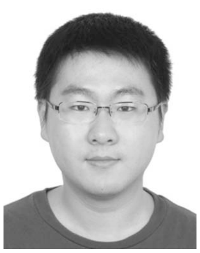
[29] K. Yamanishi and Y. Maruyama, "Dynamic syslog mining for network failure monitoring," in *Proc. 11th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining (KDD)*, New York, NY, USA, 2005, pp. 499–508.

[30] B. Yan and G. Chen, "AppJoy: Personalized mobile application discovery," in *Proc. 9th Int. Conf. Mobile Syst. Appl. Serv. (MobiSys)*, New York, NY, USA, 2011, pp. 113–126.

[31] H. Zhu, E. Chen, H. Xiong, H. Cao, and J. Tian, "Mobile App classification with enriched contextual information," *IEEE Trans. Mobile Comput.*, vol. 13, no. 7, pp. 1550–1563, Jul. 2014.

[32] H. Zhu *et al.*, "Mining personal context-aware preferences for mobile users," in *Proc. IEEE 12th Int. Conf. Data Mining (ICDM)*, Brussels, Belgium, 2012, pp. 1212–1217.

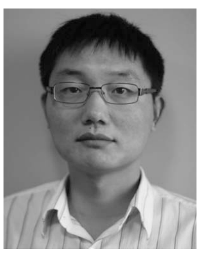
[33] H. Zhu, H. Xiong, Y. Ge, and E. Chen, "Ranking fraud detection for mobile Apps: A holistic view," in *Proc. 22nd ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, San Francisco, CA, USA, 2013, pp. 619–628.



Hengshu Zhu received the B.E. degree in computer science from the University of Science and Technology of China, Hefei, China, in 2009, where he is currently pursuing the Ph.D. degree with the School of Computer Science and Technology.

He was supported by the China Scholarship Council as a Visiting Research Student at Rutgers, the State University of New Jersey, Newark, NJ, USA, for one year. He has published a number of papers in refereed journals and conference proceedings, such as the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, the IEEE TRANSACTIONS ON MOBILE COMPUTING, *ACM Transactions on Intelligent Systems and Technology*, *ACM Transactions on Knowledge Discovery from Data*, ACM SIGKDD, and IEEE ICDM. His current research interests include data mining, machine learning, and social network analysis.

Dr. Zhu was the recipient of KSEM-2011 and WAIM-2013 Best Student Paper Award, during his Ph.D. thesis. He has also served as a reviewer for numerous journals, such as the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, PART B: CYBERNETICS, *Knowledge and Information Systems*, and *World Wide Web.*.



Chuanren Liu received the B.S. degree from the University of Science and Technology of China, Hefei, China, and the M.S. degree from the Beijing University of Aeronautics and Astronautics, Beijing, China, both in mathematics. He is currently pursuing the Ph.D. degree from the Management Science and Information Systems Department, Business School of Rutgers, the State University of New Jersey, Newark, NJ, USA.

His current research interests include data mining and knowledge discovery, and their applications in business analytics. He has published several papers in refereed journals and conference proceedings, such as *Knowledge and Information Systems*, IEEE ICDM, SIAM SDM, and ACM SIGKDD.

Yong Ge received the B.E. degree in information engineering from Xi'an Jiao Tong University, Xi'an, China, in 2005, the M.S. degree in signal and information processing from the University of Science and Technology of China, Hefei, China, in 2008, and the Ph.D. degree in information technology from Rutgers, the State University of New Jersey, Newark, NJ, USA, in 2013.

He is currently an Assistant Professor with the University of North Carolina at Charlotte, Charlotte, NC, USA. His current research interests include data mining and business analytics. He has published prolifically in refereed journals and conference proceedings, such as the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, *ACM Transactions on Information Systems, ACM Transactions on Knowledge Discovery from Data, ACM Transactions on Intelligent Systems and Technology*, ACM SIGKDD, SIAM SDM, IEEE ICDM, and ACM RecSys.

Prof. Ge was the recipient of the ICDM-2011 Best Research Paper Award, the Dissertation Fellowship at Rutgers University, in 2012, and the Excellence in Academic Research (one per school) at Rutgers Business School, in 2013. He has served as a Program Committee Member at the ACM SIGKDD, in 2013, the International Conference on Web-Age Information Management, in 2013, and IEEE ICDM, in 2013. He was a reviewer for numerous journals, including the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, PART B: CYBERNETICS *ACM Transactions on Intelligent Systems and Technology, Knowledge and Information Systems*, and *Information Science*.

Hui Xiong (SM'07) received the B.E. degree in automation from the University of Science and Technology of China, Hefei, China, the M.S. degree in computer science from the National University of Singapore, Singapore, and the Ph.D. degree in computer science from the University of Minnesota, Minneapolis, MN, USA.

He is currently a Professor and a Vice Chair of the Management Science and Information Systems Department, and the Director of Rutgers Center for Information Assurance with Rutgers, the State University of New Jersey, Newark, NJ, USA, where he received a two-year early promotion/tenure in 2009. His current research interests include data and knowledge engineering, with a focus on developing effective and efficient data analysis techniques for emerging data intensive applications. He has published three books, over 40 journal papers, and over 60 conference papers.

Prof. Xiong was the recipient of the Rutgers University Board of Trustees Research Fellowship for Scholarly Excellence in 2009 and the ICDM-2011 Best Research Paper Award from Rutgers, the State University of New Jersey. He is a Co-Editor-in-Chief of Encyclopedia of GIS and an Associate Editor of the IEEE TRANSACTIONS ON DATA AND KNOWLEDGE ENGINEERING and *Knowledge and Information Systems* journal. He has served regularly on the organization and program committees of numerous conferences, including as a Program Co-Chair of the industrial and government track for the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining and for the IEEE International Conference on Data Mining (ICDM-2013).

Enhong Chen (SM'07) received the B.S. degree from Anhui University, Hefei, China, the master's degree from the Hefei University of Technology, Hefei, and the Ph.D. degree in computer science from the University of Science and Technology of China (USTC), Hefei.

He is currently a Professor and a Vice Dean with the School of Computer Science and a Vice Director of the National Engineering Laboratory for Speech and Language Information Processing, USTC. His current research interests include data mining and machine learning, social network analysis, and recommender systems. He has published several papers in refereed journals and conferences, including the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, the IEEE TRANSACTIONS ON MOBILE COMPUTING, KDD, ICDM, NIPS, and CIKM.

Prof. Chen is the recipient of the National Science Fund for Distinguished Young Scholars of China. He was the recipient of the Best Application Paper Award on KDD-2008 and the Best Research Paper Award on ICDM-2011. He has served on program committees of numerous conferences, including KDD, ICDM, and SDM.