Understanding Smartphone Users From Installed App Lists Using Boolean Matrix Factorization

Sha Zhao, Gang Pan, *Member, IEEE*, Jianrong Tao, Zhiling Luo, Shijian Li, and Zhaohui Wu, *Fellow, IEEE*

***Abstract*—Smartphones are changing humans' lifestyles. Mobile applications (apps) on smartphones serve as entries for users to access a wide range of services in our daily lives. The apps installed on one's smartphone convey lots of personal information, such as demographics, interests, and needs. This provides a new lens to understand smartphone users. However, it is difficult to compactly characterize a user with his/her installed app list. In this article, a user representation framework is proposed, where we model the underlying relations between apps and users with Boolean matrix factorization (BMF). It builds a compact user subspace by discovering basic components from installed app lists. Each basic component encapsulates a semantic interpretation of a series of special-purpose apps, which is a reflection of user needs and interests. Each user is represented by a linear combination of the semantic basic components. With this user representation framework, we use supervised and unsupervised learning methods to understand users, including mining user attributes, discovering user groups, and labeling semantic tags to users. Extensive experiments were conducted on three data subsets from a large-scale real-world dataset for evaluation, each consisting of installed app lists from over 10 000 users. The results demonstrated the effectiveness of our user representation framework.**

***Index Terms*—Boolean matrix factorization (BMF), installed app lists, smartphones, user attributes.**

I. Introduction

SMARTPHONES have increasingly become an indispensable part in our daily lives. Smartphones serve a wide variety of functions, and users can exploit mobile applications to achieve many imaginable purposes. The mobile application market has seen explosive growth in recent years, with Apple's app store having around 1.8 million applications and Google's Android market also having about 2.5 million applications as of the third quarter of 2019.[1] Applications (Apps) on smartphones can be considered as entries to access everyday life services, such as communication, shopping, and navigation. Smartphone users install apps depending on their needs, preferences, and tastes. Since a smartphone is linked to an individual, the apps on it achieve greater personalization. Thus, installed app lists can effectively convey lots of personal information. This provides us with a new lens to understand users from their installed app lists on smartphones.

User understanding is a process of identifying user characteristics from any related data, to have knowledge of what they look like, what they need, their abilities, limitations, etc. This knowledge can be further used for enhancing the retrieval for providing satisfaction to users in the context of devices, services, and applications. In particular, devices could adjust automatically in a smart environment depending on users' needs and preferences. Devices could be targeted toward improving the user experience of specific users by providing more valuable features than others. Services can be actively recommended to users accordingly. Knowing user personal information can also be leveraged to enhance the personalization of applications, such as personalized Web search, personalized recommendation, and targeted advertising.

Besides, it can help users understand themselves in an objective and extensive way, so as to improve life quality. Users' behavioral observations are surprisingly weakly related to their cognitive reports [1], [2]. Behaviors recorded by smartphones can help discover objective and unobservable information about users themselves. In addition, people's memory capacity is limited, while smartphone records are infinite and detailed. Smartphones can continuously collect records about user behaviors for a long duration and in detail, which is helpful for extensively understanding users.

As with the prevalence of mobile apps in recent years, understanding users using smartphone apps has been emerging as a new research field. A growing number of studies have sought to understand mobile users from smartphone apps, such as demographics [3]–[5], personality [6], and daily life [7], [8]. The features used in these studies, however, are relatively simple and straightforward, which cannot compactly represent smartphone users' characteristics.

In this article, in order to understand smartphone users, we propose a user representation framework to build a compact and semantic feature space from installed app lists using

Manuscript received October 24, 2019; revised December 27, 2019; accepted January 7, 2020. This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFC1504006, in part by the Natural Science Foundation of China under Grant 61772460, Grant 61802340, Grant 61802342, and Grant 61925603, and in part by the China Postdoctoral Science Foundation under Grant 2017M620246 and Grant 2018T110591. This article was recommended by Associate Editor S. Cruces. *(Corresponding author: Gang Pan.)*

Sha Zhao, Gang Pan, Zhiling Luo, Shijian Li, and Zhaohui Wu are with the Department of Computer Science, Zhejiang University, Hangzhou 310027, China (e-mail: szhao@zju.edu.cn; gpan@zju.edu.cn; luozhiling@zju.edu.cn; shijianli@zju.edu.cn; wzh@zju.edu.cn).

Jianrong Tao was with the Department of Computer Science, Zhejiang University, Hangzhou 310027, China. He is now with Fuxi AI Lab, NetEase, Inc., Hangzhou 310052, China (e-mail: hztaojianrong@corp.netease.com).

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

Digital Object Identifier 10.1109/TCYB.2020.2967644

[1] https://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/

Boolean matrix factorization (BMF). With this feature space, we identify user characteristics at both an individual user level and a user group level, such as mining user attributes, discovering user groups, and labeling semantic tags to users. The contributions of this article are two-fold.

1) We propose a new user representation framework from installed app lists, where we exploit the underlying relations between apps and users using BMF. It builds a compact user subspace by discovering basic components from installed app lists. Each basic component encapsulates a semantic interpretation of a series of special-purpose apps, so that it is easy to interpret. Each user is represented by a linear combination of the basic components, reflecting user needs or interests.
2) With the user representation framework, we develop three typical application scenarios to understand users. More specifically, at an individual user level, we mine three user attributes with classification, achieving an average accuracy of 84%. We also extract semantic labels for each individual. At a user group level, we discover user groups using clustering and identify the characteristics of each user group. We evaluate the effectiveness using three data subsets from a large-scale real-world dataset, each consisting of installed app lists of more than 10 000 smartphone users.

II. Related Work

Recently, a growing number of analyses have sought to understand users from various cues, such as word use [9]; audio signals [10]; Web search logs [11], [12]; and social network [13]. Compared with the cues, apps on smartphones are inclined to be more personalized, since a smartphone is not only possessed by the same user but also goes almost everywhere with the owner. This promotes research studies on understanding users from smartphone apps. Here, we will review the related work in three aspects: 1) inferring demographics; 2) explaining personality; and 3) discovering life patterns.

A. Inferring Demographics

Apps on smartphones were used to infer demographics [3]–[5], [14]–[17]. For example, Seneviratne *et al.* [3] inferred about 200 users gender from their installed app lists, with an accuracy around 70%. Qin *et al.* [5] inferred gender and age range by leveraging the differences on app usage behaviors of 32 660 users, with an accuracy of 81.12% and 73.84%, respectively. Malmi and Weber [15] analyzed the used app lists of 3760 Android users, and inferred gender and income using logistic regression (LR) with an accuracy of 82.3% and 60.3%, respectively. Zhao *et al.* [4] mined 12 user attributes from installed app lists by using the support vector machine (SVM), with an average equal error rate of 16%. It was shown that user attributes have a significant impact on the adoption of apps.

B. Explaining Personality

The correlation between one's personality and his/her app usage behaviors has been analyzed [6], [14]. For example, Chittaranjan *et al.* [6] investigated the relationship between app usage behaviors derived from rich smartphone data and self-reported Big-Five personality traits (Extraversion, Agreeableness, Conscientiousness, Neuroticism, and Openness to Experience). They analyzed the app usage records on the Nokia 95 smartphone from 83 participants over eight months, and found that the usage of all applications, except the use of Maps, Camera, Chat, and Game applications significantly explained variance in the traits. They also classified users' Big-Five traits with an accuracy of 75.9%. Xu *et al.* [14] also explained the adoption of thirteen mobile apps by using the Big Five personality traits. For example, conscientiousness has a negative and significant effect on the adoption of services like photography, media and video, and location-based services.

C. Discovering Life Patterns

There have been a few studies trying to infer life patterns from smartphone apps [7], [18], [19]. In [7], life stages, such as "single," "couple without children," and "family," were predicted with an accuracy of 85%, based on the adoption of apps of 1453 users. It was found that the adoption behaviors of apps differ across different life stages, since users have different needs. Zhao *et al.* [18] discovered different user groups by analyzing the app usage behaviors from 106 672 Android users, and identified the characteristics of each user group. For example, one user group uses the apps of Phone and SMS during midnight much more frequently than the others.

Although a few studies have used apps on smartphones to understand users, the features used in the studies are simple and straightforward. They cannot well describe the complex relationships between apps and users, degrading the performance of understanding users. In this article, we propose a new user representation framework where we explore the underlying relations between apps and users using the method of BMF, to build a compact and semantic feature space.

III. User Representation Framework With Installed App Lists

A. Motivation

To understand users from their installed app lists, we need to extract valuable features to represent each user. Intuitively, apps can be used for user representation. If an app is installed by one user, it appears *only once* in the installed app list even though it is updated multiple times by the user. If not, the app does not appear. Thus, we exploit the Boolean nature of the installed app list to represent each user as a Boolean app-based vector, where each app serves as a dimension, and each dimension has two values, 1 and 0, indicating whether the app is installed. The app-based vector directly describes whether one user installs one app, however, there are some limitations if we directly use each app to represent one user.

First, one single app cannot comprehensively and accurately convey a certain user characteristic. On one hand, only one app is not enough to describe a semantic. For example, only Alipay (a payment app) cannot reflect the preference for online shopping, since Alipay can also be used for payment for offline shopping. If one user installs both Taobao (an online shopping app) and Alipay, it can indicate this user prefers to shop online on his/her smartphone. On the other hand, one app sometimes indicates different user characteristics. For example, BeautyShopping (Meiligou) is an app designed for fashion shopping targeted for young ladies, and could reflect the preference to online shopping, age, and gender to some extent. It is necessary to combine other apps to more accurately and semantically infer the user characteristics.

Second, there would be a dimensionality curse and redundancy issue. A user vector would be very long if all the apps are used for user representation. Besides, in an installed app list, some apps are redundant for describing one user. Actually, some different apps that are interrelated with each other serve for similar tasks, so that they can reflect the same user characteristics. We roughly summarize the interrelationship between apps into the following types.

1) *Being Similar in Function and Providing Similar Services:* For example, both of IQiYi video and Youku video can reflect that one likes watching videos with smartphones, since both of them provide online video services.
2) *Being Associated in Function and Usually Installed Together to Serve One Need:* For example, Taobao and Alipay usually appear together on a smartphone, since users often use Taobao, an electronic commerce platform, for online shopping and Alipay for payment. It can reflect that the users prefer to online shopping on smartphones.
3) *Being Installed in Group to Provide Services for Specific User Groups:* For example, Baoyi map, Baoyi reading, Baoyi music, and Baoyi call are usually installed in the bundle, which provide a series of services for users with poor eyesight. It indicates that the users are likely visually impaired.

Taking together, we need to build a compact and semantic user subspace by discovering which apps are interrelated from installed app lists. Matrix factorization can discover the interrelationship underlying the interactions between two kinds of entities [20]. It represents a data matrix as a product of two factor matrices: one containing basis vectors that represent meaningful subjects in the dataset and another describing how the observations can be expressed as combinations of the subjects. There have been some matrix factorization methods, such as single value decomposition (SVD) [21] and non-negative matrix factorization (NMF) [22].

B. BMF-Based User Subspace

As mentioned above, a user can be intuitively represented as a Boolean vector, where each dimension represents one app, and has two values, 1 and 0, indicating whether the app is installed or not. Formally, a user u_i is represented as $u_i = (x_1, x_2, \ldots, x_m)$, where x_j is the jth app. Thus, all the user vectors construct a Boolean user matrix U, and we can discover the interrelated apps from it by matrix factorization methods to build a compact and semantic user subspace.

Some matrix factorization methods allow the factor matrices to contain arbitrary real numbers (e.g., NMF) or even negative entries (e.g., SVD). However, it is hard to interpret real-valued factor matrices if the input matrix is binary. Thus, it is natural to require that the factor matrices are also binary. Considering the user matrix U is binary, we exploit a method of BMF.

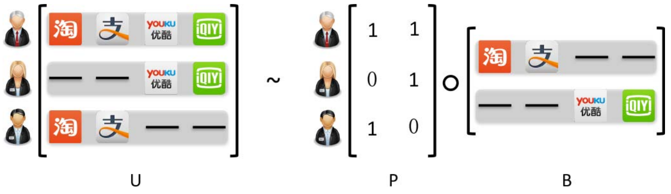


Fig. 1. Three-user example of BMF.

Definition 1 (BMF-Based User Subspace): Given the binary $n \times m$ user-app matrix U where n rows represent n users and m columns represent m apps, and a positive integer k, find an $n \times k$ binary matrix P and a $k \times m$ binary matrix B that minimize

$$|U - P \circ B| = \sum_{i=1}^{n} u_i \oplus (P \circ B)_i \tag{1}$$

where $\circ$ denote the Boolean product, that is, the matrix product with addition defined by $1 + 1 = 1$, and $\oplus$ means a bitwise exclusive OR.

We denote a row vector of a matrix M by m_i, a column vector by $m_{.i}$, and a matrix entry by m_{ij}. Formally, BMF can be formulated by

$$\begin{bmatrix} u_1 \\ \cdots \\ u_i \\ \cdots \\ u_n \end{bmatrix} \approx \begin{bmatrix} u'_1 \\ \cdots \\ u'_i \\ \cdots \\ u'_n \end{bmatrix} = \begin{bmatrix} p_1 \\ \cdots \\ p_i \\ \cdots \\ p_n \end{bmatrix} \circ \begin{bmatrix} b_1 \\ \cdots \\ b_j \\ \cdots \\ b_k \end{bmatrix} \tag{2}$$

where $\{u'_i\}_{i=1}^{n}$ refers to U', $\{p_i\}_{i=1}^{n}$ refers to P, and $\{b_j\}_{j=1}^{k}$ refers to B. $U' = P \circ B$ denote the Boolean product of P and B.

The factor matrix B contains basic components that encapsulate semantic interpretation of the interrelated apps, and P describes how users can be expressed as combinations of the basic components. In B, each row vector b_i represents the ith basic component, containing information about which apps appear. $b_{ij} = 1$ if the jth app appears in the ith basic component, and $b_{ij} = 0$ otherwise. In the matrix P, each row vector p_i describes how the ith user can be represented by a linear combination of the basic components. $p_{ij} = 1$ denotes the ith user has the jth basic component. The basic components in B are used to represent each user, and B is defined as the *subspace matrix*. P can be taken as users' coordinate in the user subspace, defined as the *user coefficient matrix*. The user u_i can be represented as p_i with the user subspace.

With the factor matrices, each user u'_i can be expressed by

$$u'_i = p_i \circ B \tag{3}$$

u'_i is the logical OR of the rows of B for which the corresponding entry in the ith row of P is 1.

Fig. 1 shows an example of BMF-based user subspace. In particular, let U be a 3 × 4 Boolean matrix, where the rows represent the users u_x, u_y, and u_z, and the columns denote four apps, Taobao, Alipay, Youku video, and IQIYI video. For this

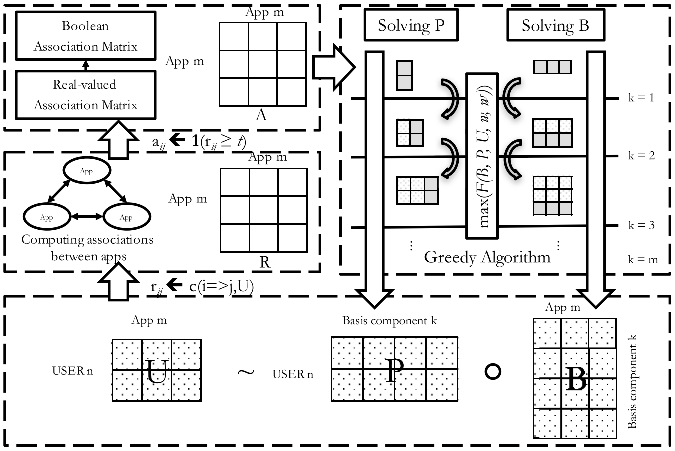

Boolean Association Matrix
Real-valued Association Matrix
App m
App m
A
$a_{ij} \leftarrow \mathbf{1}(r_{ij} \geq t)$
App
App
App
Computing associations between apps
App m
App m
R
$r_{ij} \leftarrow c(i{=}{>}j,U)$
Solving P
Solving B
max(F(B, P, U, w, w))
k = 1
k = 2
k = 3
k = m
Greedy Algorithm
App m
USER n
U
~
Basis component k
USER n
P
○
App m
B
Basis component k

Fig. 2. Algorithm of app list-based BMF.

Boolean matrix, BMF produces the representation with $k = 2$, and generates a 3×2 coefficient matrix and a 2×4 subspace matrix.

As shown in Fig. 1, the representation has no error and is easy to interpret. The columns of P assign users to different basic components and the rows of B define the apps required in each basic component. In the *subspace matrix B*, there are two basic components. The first one consists of Taobao and Alipay, reflecting the preference for online shopping. The second basic component consits of Youku video and IQIYI video, indicating that users like watching online videos on their smartphones. As we can see from the *user coefficient matrix P*, u_x has both of the basic components and may prefer to online shopping and watch videos on smartphones; u_y is with the second basic component, who likes watching videos on smartphones; and u_z has the first basic component and prefers to online shopping on his/her smartphone.

C. Solving the User Subspace

The user subspace built via BMF is compact and easy to interpret, however, it is hard to find an exact decomposition of the user matrix U. Instead, we try to find an approximate decomposition of U that minimizes the representation error. Assuming given a candidate matrix A, we can construct P column by column by selecting greedily all the rows of A as the rows of B. Here, we apply the greedy algorithm called ASSO [23]. The basic idea of ASSO is to exploit the correlation between the app columns in the user-app matrix U and build candidate basic vectors for basic components selection. First, the associations between every two app columns are computed and form an $m \times m$ real-valued matrix R. Second, the associations are turned to Boolean values to form an *association matrix A*, where each row is considered as a candidate for being a basic component. Then, a small set of candidate basic components are selected from A in a greedy way to form B, and P was fixed at the same time. The algorithm is illustrated in Fig. 2, and the pseudocode is summarized in Algorithm 1.

We first construct the association matrix A following the lines from 2 to 5 in Algorithm 1. We compute the confidence of an association r_{ij} $(0 \leq r_{ij} \leq 1)$ between the ith app and jth app via association rule mining [24], shown in (4), and we obtain an $m \times m$ real-valued matrix R. In order to transform the R into a Boolean association matrix, a threshold t is introduced to control the level of confidence required to include an app to the basic component candidate. An association between app i and app j is $t-strong$ if $r_{ij} \geq t$. We set $a_{ij} = 1$ if $r_{ij} \geq t$, and otherwise $a_{ij} = 0$. Here, the indicator function $\mathbf{1}(X)$ (in the line 3 in Algorithm 1) takes a value of 1 if proposition X is true and 0 otherwise. By doing so, a Boolean association matrix A is constructed, in which each row is a basic component candidate

$$r_{ij} = u(i \Rightarrow j, U) = \langle u_{.i}, u_{.j}\rangle / \langle u_{.i}, u_{.i}\rangle \tag{4}$$

where $\langle\cdot,\cdot\rangle$ is the vector inner product operation.

Algorithm 1 Algorithm for BMF

Input: A matrix $U \in \{0,1\}^{n\times m}$ for the user-app matrix, a positive integer k, a threshold value $t \in (0,1]$, and real-valued weights w and w^-.
Output: Matrices $B \in \{0,1\}^{k\times m}$ and $P \in \{0,1\}^{n\times k}$.

```
 1: function ASSO($U, k, t, w, w^-$)
 2:     for $i = 1, \ldots, m$ do
 3:         $a_i \leftarrow (\mathbf{1}((u(i \Rightarrow j, U)) \geq t))_{j=1}^{m}$
 4:         // Construct the association matrix $A$ row by row
 5:     end for
 6:     $B \leftarrow [\,], P \leftarrow [\,]$
 7:     // $B$ and $P$ are empty matrices.
 8:     for $l = 1, \ldots, k$ do
 9:         // Select the $k$ vectors from $A$.
10:         $(a_i, p) \leftarrow \arg\max_{a_i, p \in \{0,1\}^{n\times 1}}$
11:         $F\left(\begin{bmatrix} B \\ a_i \end{bmatrix}, \begin{bmatrix} P\ p \end{bmatrix}, U, w, w^-\right)$
12:         $B \leftarrow \begin{bmatrix} B \\ a_i \end{bmatrix}, P \leftarrow \begin{bmatrix} P\ p \end{bmatrix}$
13:     end for
14:    return $B$ and $P$
15: end function
```

Then, following the lines from 6 to 14 in Algorithm 1, *the factor matrices B and P are formed in a greedy manner.* Initially, P and B are empty matrices (line 6 in Algorithm 1). The basic components are selected from A to fix $B$and the columns of P are fixed greedily as follows: B is updated in the iteration l by adding the lth row b_l which is a row vector from A, and matrix P is updated by adding the lth column $p_{.l}$ which is an arbitrary n-dimensional binary column vector. In each iteration, in order to control the error, the objective function shown in (1) penalizes for both types of errors: 1) 0 becoming 1 in the reconstruction and 2) 1 becoming 0. We introduce weights w and w^- to reward for covering 1s and penalize for covering 0s, respectively. The selection of b_l and $p_{.l}$ is done so to maximize $F(B, P, U, w, w^-)$ formulated as (5), the value of which can be considered as the "profit" of describing U using matrices B and P. In F, when more 1s keep as 1s in the reconstruction, there are fewer 1s becoming 0s, since the number of 1s in U is constant. Similarly, when fewer 0s

TABLE I
ILLUSTRATION OF SIX BASIC COMPONENTS

No.	Apps in each basis component	Main functions	Interrelationship among apps	Attribute reflected by each basis component
1	Mogujie Meilishuo Jumeiyoumin	Social commerce fashion apps targeting females	Similar in function	Preference to online fashion shopping
2	Kugou music Netease music Tiantian music QQ music	Playing music	Similar in function	Interests in listening music with phones
3	Dazhongdianping	Group-buying	Associated in function	Preference to online shopping
	Meituan	Group-buying		
	Alipay	Online payment		
4	LOL	Game: League of legend	Associated in function	Interests in playing LOL
	LOL box	A game assistant for LOL		
5	Baby learning numbers Help-Baby learning pinyin Help-Baby learning fruit Help-Baby learning animals	Help babies learn numbers, language, fruit, and animals	Installed in group	parents or raising a baby
6	Meituxiuxiu	Beautifying pictures	Installed in group	Preferences-to beautifying pictures and videos
	Meiyan Camera	BeautyPlus-Selfie Camera		
	Meipai MV	BeautyPlus-recording short videos		

become 1s after reconstruction, there are more 0s kept as 0s

$$F = w|\{(i,j):u_{ij} = 1, (P \circ B)_{ij} = 1\}| - w^{-}|\{(i,j):u_{ij} = 0, (P \circ B)_{ij} = 1\}|. \quad (5)$$

In this way, the user subspace can be built. The association matrix A is constructed in time $O(nm^2)$, and a single discrete basic component is obtained in time $O(nm^2)$. There are k iterations to fix the factor matrices of B and P. Thus, Algorithm 1 has time complexity $O(knm^2)$. In Algorithm 1, there are two parameters: 1) the threshold t and 2) weight w (assuming that $w^{-} = 1$) controlling the quality of the results. t controls the number of apps appearing in each basic component, and the weight w can be used to control the performance of BMF.

D. Illustration of Basic Components

The basic components discovered by BMF encapsulate semantic interpretation of interrelated apps. To give a sense of the basic components, we list six examples in Table I, including the apps in each basic component, the main functions of each app, the interrelationship between apps, and the user attribute reflected by each basic component.

As shown in Table I, the first two basic components consist of apps that are similar in function. For example, the first one consists of three apps that are social commerce fashion apps targeting females, and the second one consists of the apps providing services for playing music. They reflect the user attributes of the preference to online fashion shopping and listening to music on smartphones, respectively. The third and fourth basic components are composed of apps associated in function. In the fourth basic component, there is a game of league of legend (LOL) and LOL assistant box. The users with this component like playing the game of LOL on their smartphones. The last two basic components consist of apps that are usually installed in the group. For example, the fifth component consists of apps that are for babies learning, with which the users are probably raising a baby.

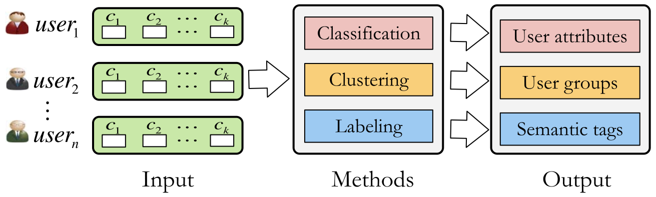


Fig. 3. Diagram of understanding users in the user subspace (c_k: the kth basic component).

IV. UNDERSTANDING USERS IN BMF-BASED USER SUBSPACE

Given a user population with installed app lists, a BMF-based user subspace B can be built by the method of BMF. One user can be basically described by his/her installed app list as $u_i = (x_1, x_2, \ldots, x_m)$, where x_j is for the jth app, and it has two values, 1 and 0, for indicating whether the app is installed or not. With the BMF-based user subspace B, the user can be approximately represented by

$$p_i = u_i \circ B^{\dagger} \quad (6)$$

where $B^{\dagger}$ is the pseudoinverse matrix of B.

With the user representation, we develop three typical application scenarios to understand users: 1) mining user attributes with classification; 2) discovering user groups with clustering; and 3) labeling users with extracted semantic tags. Fig. 3 shows the diagram of understanding users with the user subspace.

A. Mining User Attributes With Classification

User attributes can be roughly divided into two classes: 1) discrete attributes and 2) continuous attributes. The former refers to the ones with discrete values such as age, while the latter are with continuous values such as height. The exact values of attributes are not so necessary in some application scenarios, such as targeted advertising and personalized recommendation. Thus, the values of attributes can be discretized so that the attributes can be divided into z categories with z corresponding labels. For example, height could be divided into three classes, short, medium, and tall. Thus, the problem of mining user attributes can be simplified to identifying which class a user belongs to, and what label he/she has. From the viewpoint of classification, it is a multiclassification problem [25]. It can be defined as: given an attribute α and its label collection $L = \{l_1, l_2, \ldots, l_z\}$, for a user p_i represented in the BMF-based user subspace, find a function

$$y(\alpha) : p_i \rightarrow l_j \quad (7)$$

where p_i is the input of the classifier, and $l_j \in L_i$ is the output.

B. Discovering User Groups With Clustering

A user group is a set of people who have similar attributes, such as interests and needs. In the real-world, user attributes could shape their adoption of smartphone apps, and users with similar attributes may install similar apps, attempting to aggregate into a group. In our compact user subspace, users with similar basic components discovered from apps form a cluster.

In the case of clustering, it is to group a given collection of unlabeled patterns into meaningful clusters based on similarity [26], [27]. Therefore, the task of discovering user groups is smoothly transformed into a clustering problem. It can be solved by segmenting users into clusters, with the most similar users being grouped into the same one cluster. In other words, whereas a user in a certain group should be as similar as possible to all the other users in the same group, it should likewise be as distinct as possible from users in different groups. Given a user population $\{p_1, p_2, \ldots, p_n\}$, we aim to partition the n users into q $(q \leq n)$ groups $G = \{G_1, G_2, \ldots, G_q\}$ that minimize

$$\arg\min_{G} \sum_{i=1}^{q} \sum_{p \epsilon G_i} \|p - \mu_i\|^2 \tag{8}$$

where μ_i is the mean of the users in the group G_i.

C. Labeling Users With Semantics of Basic Components

Labeling each user p_i with semantic tags is helpful for well-understanding users, which can describe users in a brief and intelligible way [28]. With the BMF-based user subspace, each user can be represented as p_i with basic components. Each basic component encapsulates a semantic interpretation of a series of special-purpose apps. One user can be labeled by the semantic tags of the basic components he/she has. It is required to extract the semantic s_j of the basic component b_j, which can be retrieved from the main functions of the apps appearing in b_j. The function of apps can be derived from their description available in appstore websites. However, reliable semantic extraction from text remains difficult. We choose to avoid this problem and use an alternative in the form of crowdsourcing, by soliciting contributions from a large group of people and especially from the online community [29]. In our case, for each basic component b_j, participants are asked to select a semantic word from the candidate word set we list for b_j, depending on their knowledge to the apps appearing in b_j. We gather the words, and select the most frequent word as the semantic tag s_j of the basic component b_j.

Then, one user p_i can be labeled with the semantics of the basic components he/she has, described as (9). The number of one's semantic tags is equal to that of the basic components appearing in his/her representation vector, since only one semantic tag is allowed for each basic component

$$\text{Tag}(p_i) = \{s_j | p_{ij} = 1\} \tag{9}$$

where Tag(p_i) represents the collection of the tags of the user p_i, p_{ij}=1 means the user p_i has the jth basic component.

V. Experiments

In this section, we evaluated the BMF-based user subspace using real-world datasets of installed app lists. The results of mining user attributes with classification, discovering user groups with clustering, and labeling users with semantic tags were analyzed in detail, respectively.

A. Data Description

The datasets we used contain lists of apps installed on Android smartphones, provided by a mobile Internet company in China. Each record in the dataset consists of an anonymized User ID (the unique identity), installation package name used to identify an app, and the app name.

B. Getting of Attributes

For the evaluation, groundtruth of user attributes is required. A large-scale groundtruth dataset, however, is difficult to obtain. To cope with this problem, we employed two strategies to obtain three kinds of groundtruth for our experiment: 1) gender; 2) smartphone price; and 3) screen size.

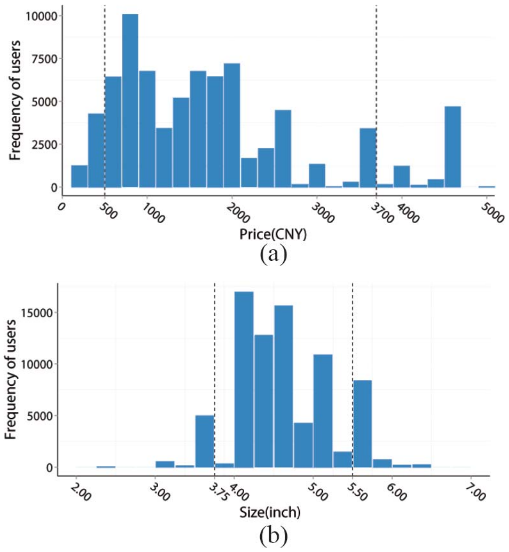
1) Getting Gender via Questionnaires: Gender information was collected through questionnaires. When the installed app lists were collected, a brief questionnaire about demographics such as gender was meanwhile present to users. Users voluntarily answered the questionnaire, and the user who reported the demographic attribute was offered a compensation with a 20 Yuan e-coupon. Then, one's hashed ID is uploaded to the server, as well as the corresponding installed app lists and gender information. There are 15 000 participants voluntarily providing their gender, including 7500 males and 7500 females.

2) Getting User Attributes via Smartphone Models: We designed the other two user attributes: 1) smartphone price and 2) smartphone size from phone model-related data. Each smartphone in our real-world dataset is accompanied with its phone model. For each smartphone, we crawled its model-related data from the Web. According to the 2017 annual report on Chinese smartphone market by iiMedia,[2] price is an important factor to be considered in the purchase of smartphones. Smartphone price can reflect user income or consumption level to a certain extent. Smartphone size is another important factor. Screen size may reveal preference of phone usage to some extent.

Smartphone Price: To give a sense of the smartphone price in the dataset, we computed the frequency of the users in terms of their smartphone price, shown in Fig. 4(a). The horizontal axis is the price of smartphones with a 200 CNY bin width. As shown, there is a wide range of the price, from 200 CNY ($32) to 6000 CNY ($968). Few users have smartphones with the price lower than 500 CNY ($75) or higher than 3700 CNY ($560). Here, we focused on those users who have low-price or high-price phones, and divided them into two groups: 1) *low-price group* with phone price lower than 500 CNY (negative samples) and 2) *high-price group* with phone price higher than 3700 CNY (positive samples). There are 6253 and 6656 users in the low- and high-price group, respectively.

Smartphone Size: Similarly, we computed the frequency of the users in terms of their screen size, shown in Fig. 4(b), where the horizontal axis is the screen size of smartphones with a 0.25-inch bin width. The smartphone screen size varies from 3.0 to 6.0 inch. Here, we focused on the users who use smartphones with size smaller than 3.75 inch and larger than

[2] http://www.iimedia.cn/56041.html



(a)

(b)

Fig. 4. (a) Frequency of smartphones in terms of price, (b) Frequency of smartphones in terms of screen size.

TABLE II
THREE DATA SUBSETS FOR EXPERIMENTS

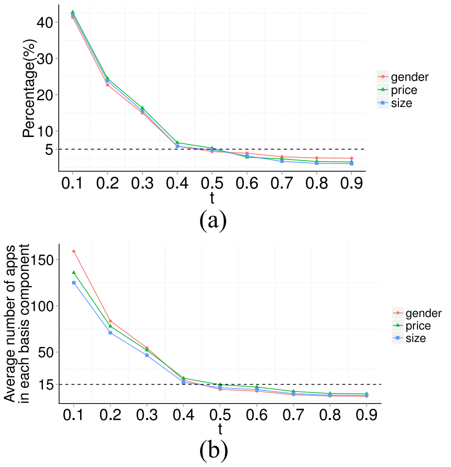
Dataset	Users	Apps		
gender	15,000	9326	male	female
			7500	7500
price	12,909	6655	high	low
			6656	6253
size	11,176	5177	large	small
			5588	5588

5.50 inch, and divided them into two groups: 1) *small-size group* with size smaller than 3.75 inch (negative samples) and 2) *large-size group* with size larger than 5.50 inch (positive samples). There are 9911 negative samples and 5588 positive samples in the dataset, respectively. In order to keep the balance of the two-class samples, we randomly selected 5588 users with screen size smaller than 3.7 inch as negative samples.

We focused on apps who were installed by many users. We removed those apps which appear less than three times in total, and did not use them to represent users. The overview of the three data subsets after filtering is shown in Table II. In the user-app matrix input for BMF, the percentage of 1s is 0.53%, 0.60%, and 0.57% for the three subdatasets of gender, price, and size, respectively.

C. Setting the Value of Parameter t

In Algorithm 1, there are two parameters: 1) the threshold t controls the number of apps in each basic component and 2) weight w (assuming that $w^{-} = 1$) impacts the factorization accuracy of BMF. We first conducted experiments to set the value of t, and then set the value of w according to classification results, as well as the value of k.



(a)

(b)

Fig. 5. When t varies, (a) percentage of nonzero elements in the real-valued matrix R that will be turned into 1s and (b) average number of apps in each basic component.

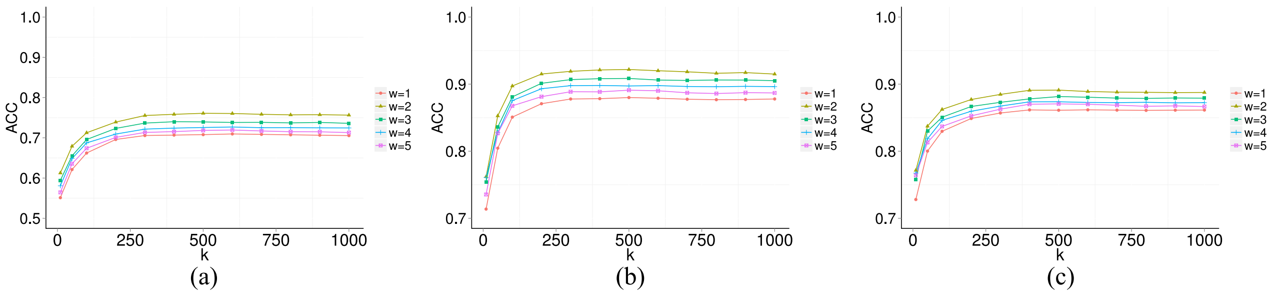
We computed the associations between every two apps and form a real-valued matrix R. t was introduced to control the level of confidence required to include an app to the basic component candidate. If the association between two apps is larger than t, the association will be turned into 1, and 0 otherwise. There are many 0s in R, indicating some apps are not interrelated with each other. Here, we focused on the associations that are stronger than in R. We computed how many nonzero elements will be turned into 1s with varying t, shown in Fig. 5(a). In Fig. 5(a), the horizontal axis is the varying t values using a 0.1 bin width, and the vertical axis is the percentage of nonzero elements that will be turned into 1s with the specific value of t. We also computed and the average number of apps in each basic component when t varies, shown in Fig. 5(b). In Fig. 5(b), the horizontal axis is the same as that of Fig. 5(a), and the vertical axis is the average number of apps in each basic component candidate with the specific value of t.

When t is smaller, there are more 1s in the association matrix A. However, if t is very small, there are too many apps in each basic component, making it difficult to interpret. For example, when t is 0.1, there are around more than 120 apps in each basic component candidate on average, shown in Fig. 5(b). On the contrary, if t is very big, there will be very few apps in each basic component, missing some interrelated apps. For example, when t is 0.9, only around three apps in each basic component candidate. Therefore, we made a tradeoff and set t =0.6, for which more than 5% of nonzero elements will be turned to 1s, and about 8, 12, and 9 apps on average appearing in each basic component candidate for gender, price, and size, respectively.

D. Classification Results

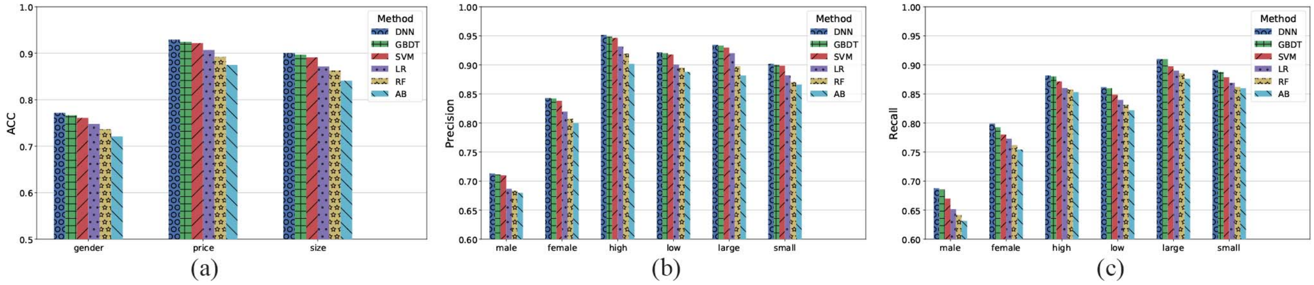
1) Performance Measurement and Implementation: We used the criterion of accuracy (abbr. *ACC*), precision, and recall to measure the classification results.

We trained different classifiers to investigate the ability of the basic components for mining user attributes, including deep neural network (DNN), gradient boosting decision tree (GBDT), SVM, LR, random forest (RF), and adaboost (AB). In particular, in the DNN model, features were input into a wide layer, followed by three hidden layers of fully connected



(a) (b) (c)

Fig. 6. Classification results with varying *w* for three user attributes, respectively. (a) Gender. (b) Price. (c) Size.



(a) (b) (c)

Fig. 7. Performance comparison among different classification methods in terms of (a) *ACC*, (b) precision, and (c) recall.

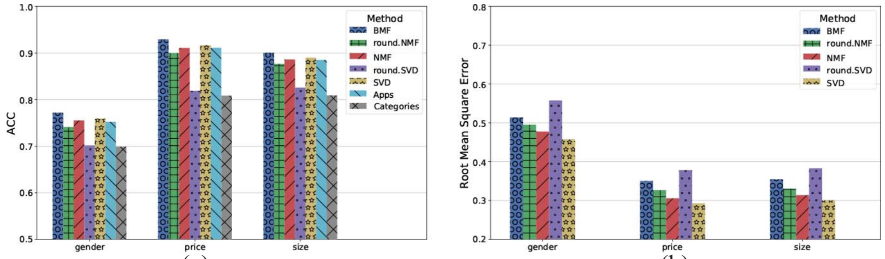
rectified linear units (ReLU). There are 64, 16, and 4 neurons on the first, second, and third hidden layer. In the training procedure, a cross-entropy loss was minimized with gradient descent on the output of the sampled softmax. To train the classifiers, we employed the five-fold cross-validation policy. The dataset was evenly divided into five folds. In each round, four folds were for training classifiers and the rest for validation. Thus, any user for testing will never simultaneously occur in the training and testing set. We repeated this procedure five times.

2) *Varying w and k for Classification:* As mentioned above, the values of weight *w* and *k* (the number of basic components discovered by BMF) can affect the classification performance. In order to decide the values of *w* and *k*, we conducted experiments with different values of *w* and *k* to observe the performance of classification for all the three data subsets, respectively, shown in Fig. 6. As we can see, for each *w*, with *k* increasing, *ACC* increases and eventually becomes steady nearly at $k = 400$ for all the three attributes. When *w* is bigger, there will be more 1s in the factor matrices. Thus, we set *k* to be 400. For each *k*, the classification performance is the best when $w = 2$ for all the three attributes, while that is the worst when $w = 1$. When the *w* is too big, it will result in severe reconstruction errors because more 0s in the original user-app matrix become 1s in the reconstruction error. The error will impact the performance of the classification tasks. The value of *w* was set to be 2.

After *t*, *w*, and *k* were decided, we also computed the percentage of 1s in the user coefficient matrix where each row represents each user for classification. The percentage of 1s in the user coefficient matrix is 7.16%, 8.18%, and 7.69% for the three attributes of gender, price, and size, respectively. The user representation vector is more compact than that in the input user-app matrix. The density has been improved around 14 times for all the three subdatasets of gender (7.16% *versus* 0.53%), price (8.18% *versus* 0.60%), and size (7.69% *versus* 0.57%). The significant density improvement shows the effectiveness of the BMF method in our datasets.

3) *Analysis of Classification Results:* In order to demonstrate the efficiency of basic components for classifying users, we conducted an extensive comparison of classification results from different aspects, including classifiers, matrix factorization methods, and user representation methods.

a) Comparison among classification methods: We tested the classification results of different classifiers (DNN, GBDT, SVM, LR, RF, and AB). We implemented all the algorithms in python, and tried different parameters in practice to achieve the best performance, respectively. For all the classifiers, each user was represented by the 400 basic components discovered by BMF with $t = 0.6$ and $w = 2$. The comparison performance of *ACC*, precision and recall is shown in Fig. 7. We can see that DNN performs best for all of the three attributes, and AB performs worst (DNN > GBDT > SVM > LR > RF > AB). For gender, the *ACC* of the six algorithms is 77.2% (DNN), 76.6% (GBDT), 76.1% (SVM), 74.7% (LR), 73.6% (RF), and 72.1% (AB), respectively. For the smartphone price, the *ACC* is 92.9% (DNN), 92.4% (GBDT), 92.1% (SVM), 90.7% (LR), 89.2% (RF), and 87.4% (AB). For the screen size, the *ACC* of the six algorithms is 90.1% (DNN), 89.7% (GBDT), 89.1% (SVM), 87.1% (LR), 86.2% (RF), and 84.1% (AB), respectively. Similarly, for the precision and recall, the DNN performs the best while the AB performs the worst, shown in Fig. 7(b) and (c). With the DNN model, the precision of gender, price, and size is 0.69 for male and 0.80 for female, 0.88 for high price and 0.86 for low price, and 0.91 for large screen size and 0.89 for small, and the recall is 0.69 for male and 0.80 for female, 0.88 for high price and 0.86 for low, and 0.91 for large screen size and 0.89 for small.



(a) (b)

Fig. 8. Performance comparison among different matrix factorization methods: BMF, NMF, round.NMF, SVD, and round.SVD. (a) Classification accuracy. (b) Reconstruction error.

b) Comparison with other matrix factorization methods: We compared BMF with the other matrix factorization methods: SVD [21], round.SVD, NMF [22], and round.NMF. In particular, by SVD and NMF, the factor matrices of the *user coefficient matrix* and *subspace matrix* are real valued, but in round.SVD and round.NMF we rounded the factor matrices to Boolean ones. The rounding for NMF was done by taking 0.5 as the threshold and setting all values less than 0.5 to 0 and all others to 1, and for SVD taking 0 as the threshold. When we used SVD, round.SVD, NMF and round.NMF, the original matrix is boolean, and $k = 400$, that is, each user was also represented as a 400-D vector.

The comparison performance of *ACC* is shown in Fig. 8(a). As we can see, the user representation built via BMF performs best for all the three attributes while that of round.SVD performs worst, about 7%–11% lower than BMF. When the methods of NMF and SVD are used, the classification results are very close, around 1%–2% lower than those of BMF. To be specific, for the attribute of gender, the ACC for BMF, NMF, round.NMF, SVD and round.SVD is 77.2%, 75.9%, 74.1%, 76.2%, and 70.1%, respectively. For the attribute of smartphone price, the ACC for BMF, NMF, round.NMF, SVD and round.SVD is 92.9%, 91.2%, 90.0%, 91.4%, and 81.9%, respectively. The ACC for BMF, NMF, round.NMF, SVD and round.SVD, for the attribute of smartphone size, is 90.1%, 88.6%, 87.6%, 89.5%, and 82.5%, respectively.

Yet, in the three subdatasets, SVD and NMF have lower reconstruction error. We measure the reconstruction error of all the matrix factorization methods using root mean square error (RMSE), shown in Fig. 8(b). It can be seen that the SVD has the lowest reconstruction error, while the round.SVD has the highest error. For all the three subdatasets, the BMF method has a little higher reconstruction error than NMF and SVD, but lower than round.SVD.

Although the SVD and NMF have lower reconstruction error, the classification results of BMF is a little higher than those of SVD and NMF. Moreover, the basic components discovered by BMF is much more easily to be interpreted, which is very important for understanding users. The basic components discovered by SVD and NMF consist of real values, and even negative values (e.g., SVD), making it difficult to extract the semantic of each basic component.

c) Comparison with other user representation methods: We compared user representation methods for classifying users. Each user was represented by basic components discovered by BMF, which was called as *BMF* for short. We tested two other user representation methods.

1) *Apps:* Using important apps to represent users. For a given attribute, 500 important apps were selected by information gain (IG) to represent each user [4]. Each user was expressed as a binary 500-D vector, in which each dimension has two values, that is, 1 and 0, for indicating whether the app is installed or not.
2) *Categories:* Using categories of apps to represent users. We categorized apps into 29 categories [18]. Each category was used to represent a user with a 29-D vector, where the value of each dimension is the exact number of apps one user installs in the category.

Fig. 8(a) shows the comparison performance with different user representations. For all of the three user attributes, the presentation method of *BMF* performed the best for classification. More specifically, for gender, the accuracy for the three representation methods was 77.2% (*BMF*), 75.2% (*Apps*), and 69.8% (*Categories*). For the attribute of price, the accuracy is 92.9% (*Basic Components*), 91.1% (*Apps*), and 80.8% (*Categories*), respectively. For the attribute of size, the accuracy is 90.1% (*BMF*), 88.5% (*Apps*), and 80.9% (*Categories*), respectively. In particular, both of the methods of *BMF* and *Apps*, which perform much better than *Categories* in mining attributes. It is difficult to distinguish users using their app categories, due to the coarse granularity.

d) Important basic components for classification: In this experiment, we demonstrated the most important basic components for classifying different user attributes. We applied the method of IG to select the top five important basic components for gender, price and size, respectively, shown in Fig. 9. The IG is higher, the basic component is more important.

As we can see from Fig. 9(a), the most important basic component is the one consisting of the apps of MeituPS, MeiyanCamera, and MeipaiMV, all of which provide the photography services in photographs or videos such as customizing exclusive beauty style to make photographs mesmerizing. The second important basic component consists of the apps of MeituPS (for photography) and Mogujie (a fashion commercial shopping platform targeting females). The third one consists of casual game apps, and the fourth one consists of beauty shopping apps. The fifth basic component consist of apps for group buying (Dazhongdianping and Meituan) and payment (AliPay). The most important basic components for price are shown in Fig. 9(b). The top one consists of apps for

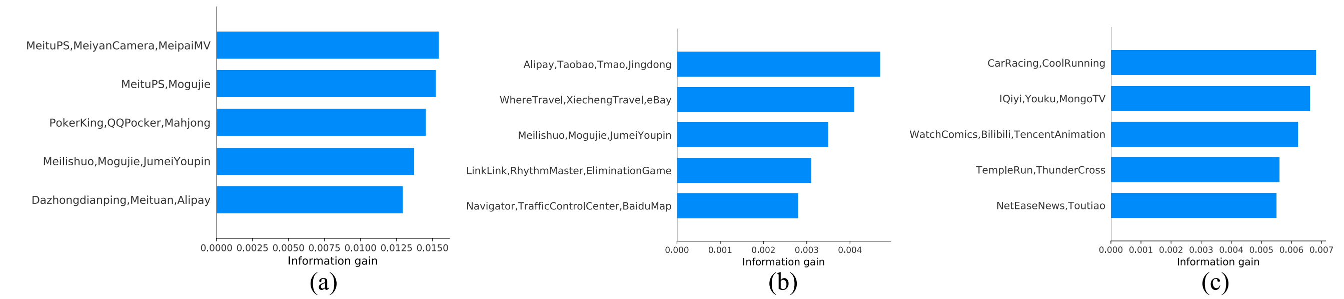


Fig. 9. Top five important basic components and the apps in each of them for (a) gender, (b) price, and (c) size.

shopping, the second one consists of apps for traveling, and the third one is composed of apps for beauty shopping. It indicates that high-price user group and low-price user group have differences in financial activities, such as shopping and traveling. As shown in Fig. 9(c), the top four important basic components are have apps for playing games and watching videos, and the fifth important one is for news reading, indicating that the users with large screen size probably like playing games, watching videos, and reading news on their smartphones.

E. Clustering Results

1) Implementation and Performance Measurement: We discovered user groups using k-means, where Euclidean distance was applied to measure the similarity among users. To obtain the optimal number of clusters, we iteratively applied k-means for a varying number of clusters and introduced γ to measure the performance. γ is the ratio of the sum square and the total sum square [30], computed by (10). Bigger γ means that data points cluster more neatly in the dimensional space

$$\gamma = \frac{BetweenSS}{TotalSS} \tag{10}$$

where *TotalSS* means the sum of squared distance of each data point in the space to the global sample mean. *BetweenSS* is the sum of squared distances of sample means to the global mean. The sample mean is the mean of each clustered group.

2) Comparing Clustering Results in Different User Subspaces: We tested user groups in two other user subspaces, *Apps-subspace*, and *Categories-subspace*, where users were represented by the methods of *Apps* and *Categories* aforementioned. We used multidimensional scaling (MDS) [31] to reduce the features to three dimensions and plotted all the clusters for all the three attributes. Fig. 10 shows an overview of the clustering results for all the three user attributes in the three different user subspaces, *BMF-based-subspace*, *Apps-subspace*, and *Categories-subspace*, respectively.

As shown in Fig. 10, for each data subset, there are no discernible patterns of clusters in the user subspaces built by important apps and app categories. Fortunately, in the BMF-based-subspace, the user groups appear to be nicely separated, giving a visual indication that *BMF-based-subspace* was successful. As for the γ, for all the three data subsets, the γ of *BMF-based-subspace* are much bigger than that of *Apps-subspace* and *Categories-subspace*. Specifically, for the gender subset, the γ of *BMF-based-subspace* is 0.75, while that of *Apps-subspace* and *Categories-subspace* is 0.44 and 0.55. For the price subset, the γ of *BMF-based-subspace*, *Apps-subspace*, and *Categories-subspace* is 0.85, 0.65, and 0.61. For the size subset, the γ of *BMF-based-subspace*, *Apps-subspace*, and *Categories-subspace* is 0.86, 0.56, and 0.63.

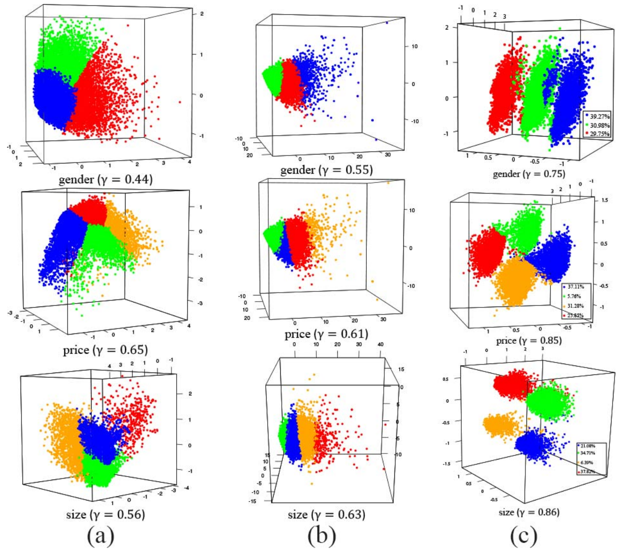


Fig. 10. Clustering results of three data subsets in different user subspaces: (a) Apps-subspace; (b) Categories-subspace; and (c) BMF-based-subspace.

3) Cluster Examples in the Gender Dataset: We took the gender subset as an example to illustrate the clustering performance in the BMF-based user subspace. In the gender subset, we obtained three clusters when the γ was the biggest. We used three different colors to differentiate the three groups, including blue, green, and red groups. For the 15 000 users in the subset, there are 39.27%, 30.98%, and 29.75% of users in the blue, green, and red group, respectively.

The users in each group are represented by combinations of similar basic components. To well understand each user group, we need to learn the top basic components of the users in each group have in common. We computed and selected the significant basic components for each user group. We compared the frequency of each basic component in one group, with its frequency in the left two groups. The significance of the ith basic component D_i was computed by (11). We ranked D_i in the descending order, and selected the top basic components for understanding each group

$$D_i = F_{i,1} - F_{i,2} = \frac{N_{i,1}}{N_1} - \frac{N_{i,2}}{N_2} \tag{11}$$

TABLE III
TOP 3 SIGNIFICANT BASIC COMPONENTS FOR EACH GROUP FOR GENDER SUBSET AND THE APPS IN THEM

Group	Top3 basis components	Apps in each basis component	Main function
Blue	1	Meitu PS (美图 PS)	Beautifying pictures
		Meiyan camera（美颜相机）	BeautyPlus-Selfie Camera
		Meipai MV（美拍视频）	BeautyPlus-recording videos
	2	Meitu PS（美图 PS）	Beautifying pictures
		Mogujie（蘑菇街）	Social commerce fashion app targeting young ladies
	3	Meilishuo（美丽说）	Social commerce fashion app targeting young ladies
		Mogujie（蘑菇街）	
		Jumei Youpin（聚美优品）	An app for group-purchasing cosmetics
Green	1	Alipay（支付宝）	An online payment platform
		Taobao（淘宝）	Apps for online shopping
		Tmao（天猫）	
		Jingdong（京东）	
	2	Dazhongdianping（大众点评）	Apps for group purchasing
		Meituan（美团）	
		Baidunuomi（百度糯米）	
	3	WhereTraveling（去哪儿）	Providing travel-booking service
		XiechengTraveling（携程旅行）	
		eBay	An app for online auction and shopping
Red	1	BeatMaster（节奏大师）	Casual games
		AngryBirds（愤怒的小鸟）	
		PlantsVsZombies（植物大战僵尸）	
	2	TempleRun（神庙逃亡）	Action games
		CoolRunning（跑酷）	
		Thunder cross（雷霆战机）	
	3	Poker King（扑克王）	Card games
		QQPoker（QQ 扑克）	
		Mahjong（麻将）	

where $F_{i,1}$ and $F_{i,2}$ are the frequency of the ith basic component for the selected group G and the other groups G^{-}, respectively. $N_{i,1}$ and $N_{i,2}$ mean the number of users with the ith basic component in G and G^{-}, respectively. N_1 and N_2 are the total number of users in G and G^{-}, respectively.

Due to space constraints, we listed the top three significant basic components for each group and the apps in them, shown in Table III. We also computed the proportion of female and male users in each group. In the blue group, female users account for 61.5%. As shown in Table III, the top three basic components for the blue group are composed of the apps providing services for beautifying pictures and videos, and group purchasing fashionable commerce and cosmetics. To be specific, the most significant basic component consists of three apps, MeituPS, Meiyan camera, and Meipai MV, which are used for beautifying pictures, taking selfie and recording videos with smart beautifying function, respectively. It indicates that, compared with others, the users in the blue group probably use apps of beautifying pictures or videos more frequently. The second most significant basic component is composed of Meitu PS, and Mogujie that is a social commerce fashion app targeting young ladies. The third most significant one consists of three apps: 1) Meilishuo; 2) Mogujie; and 3) Jumei Youpin. Meilishuo is a fashion e-commerce app similar to Meilishuo, and Jumei Youpin is an app for online group purchasing cosmetics. That is to say, the users in the blue group likely prefer to shop online for fashion commerce and cosmetics. All of these apps appearing in the top three basic components have very significant features of women.

In the green group, there are roughly equal numbers of males and females, accounting for 48.9% and 51.1%, respectively. The most significant basic component consists of four apps: 1) Taobao; 2) Tmao; 3) Jingdong; and 4) Alipay, of which the first three ones are Chinese online shopping apps similar to eBay and Amazon, and the last one is an online payment platform. The second basic component consists of Dazhongdianping, Meituan, and Baidunuomi, all of which are Chinese group buying apps for locally found consumer products and retail services. The third one consists of WhereTraveling, XiechengTraveling, and eBay, of which the first two apps provide travel-booking services. All the apps in the top three basic components provide e-commerce platforms, indicating the users in the green group prefer to online shopping.

In the red group, 66.3% of users are males. As shown in Table III, all of the apps appearing in the top three significant basic components are related to games. For example, the most significant basic component consists of apps about casual games, including BeatMaster, AngryBirds, and PlantsVSZombies. The apps in the second and third basic component are related to action games and card games, respectively. The game apps reflect the preference to play games on smartphones.

F. Labeling Results

With the BMF-based user subspace, each user is represented by basic components. We labeled each user by extracting the semantic of the basic components he/she has. The semantic of each basic component can be derived from the apps appearing in the basic component. We first investigated how many apps are in basic components. We computed the frequency of basic components in terms of number of apps with $t = 0.6$, $w = 2$, and $k = 400$. It was found that, for all the three attributes, most of basic components consist of 2 or 3 apps, accounting for about 75%, and very few basic components consist of more than 10 apps. It is reliable to manually learn the semantic of a few apps in one basic component based on the knowledge to the apps. Thus, we found out the semantic of each basic component through the approach of crowdsourcing.

1) Extracting Semantics of Basic Components Through Crowdsourcing: The semantic of each basic component was extracted through crowdsourcing. In our case, for each basic component participants were asked to select a word depending on their knowledge to the apps appearing in the basic component. Then, we selected the most frequent word as the final semantic of each basic component. More specifically, for each basic component, we manually listed three words according to the main function of the apps, as three candidate choices for participants. Participants can choose at least one word as the semantic of the basic component, or fill out other words if he/she does not agree with all the three choices.

For participants, it looked like to fill out a questionnaire based on their knowledge to the apps. There were 1200 basic components in total for the three data subsets, and we stored all the questions in a database in our local server. A simple Web page was developed to connect to the server to acquire questions. When the link of Web page was clicked by someone, 20 basic components were randomly selected to show. The Web page was published through WeChat, a very popular app for social online network in China, from July 25, 2016 to August 26, 2016. During this period, 989 people clicked our Web page, and 367 filled out the questionnaires. To ensure a high level of answer integrity, we removed the incomplete questionnaires, and there were 348 valid ones left. According to our analysis, each basic component was tagged by about five different participants to provide a level of error resistance.

For each basic component, we selected the most frequent word as its semantic. The semantics of some example basic components are given as follows.

1) *Traveling Type:* The users with this basic component like traveling. The basic component consists of apps that provide services for traveling, for example, WhereTravelling, XiechengTravelling, and RentingCar.
2) *Gamers:* The users who have the basic component are interested in playing games on their smartphones. The basic component consists of apps related to various kinds of games, for example, Fightthelandlord, Beatmaster, happyanimal, Coolrunning and Link-link.
3) *Online Shoppers:* This kind of users prefer to online shopping on smartphones. The basic component consists of Taobao and Alipay. They are associated in function, since users often use Taobao for shopping and Alipay to pay.
4) *Car Lovers:* The users are interested in car-related content. The apps providing services in car news, car forum, and car technology appear in the basic component.
5) *Parents:* The users with this basic component are probably parents or raising a baby. The basic component is composed of apps related to raising a baby, supporting pregnancy, etc., for example, HappyParenting, Babyhealth, an Fairytales.

2) Tags for Users: The basic components one user has are a reflection of what he/she needs, what he/she is interested in, etc. We used the semantics of one's basic components to tag him/her. Two examples of User A and User B are shown in Figs. 11(c) and 12(c), respectively. The User A was labeled with ten tags, such as CarLover, Gamer, MusicLover, SportsFan, etc. It indicates that he/she is fond of cars, music, games, sports, etc. The User B was also labeled with ten tags, including BeautyLover, BeautyShopper, OnlineShopper, Mother, CookingLover, MusicLover, VideoLover, GroupBuyer, TravelLover, and FitnessFan. The tags show that User B is a mother, and prefer to make herself look more beautiful. She is also interested in cooking, music, traveling, fitness, and watching online video on her smartphone. As we can see from the two examples, the semantic tags extracted from basic components can describe one user in a brief and understandable way.

In order to highlight the effectiveness of basic components in labeling users, we tested the other two methods: using 1) apps and 2) basic component names to tag users. Compared with apps and basic component names, the semantics of basic components are more meaningful for describing user characteristics. Also taking the User A and User B as examples, we used the names of apps in his/her basic components as tags, shown in Figs. 11(a) and 12(a). As we can see from Figs. 11(a) and 12(a), there are much more tags for the User A and User B than those in Figs. 11(c) and 12(c). Some tags are not easily understandable, since the names of some apps cannot directly describe their main function. Besides, some tags are redundant, because some apps can reflect the same attributes. For example, both of the apps, Autohome and Carheadline providing services in car news, technology, and markets, reflect the same user attribute of being interested in cars. All the apps of MongoTV, IQiYi, and YoukuVideo reflect the same interest of watching online video on smartphones. To summarize, it is difficult to quickly capture the user preference or interests when he/she is labeled with the names of the apps in his/her basic components.

We also took each basic component name as a tag, which was expressed by the names of the apps in each basic component. The User A and User B were taken as examples, and their tags are shown in Figs. 11(c) and 12(b), respectively. It

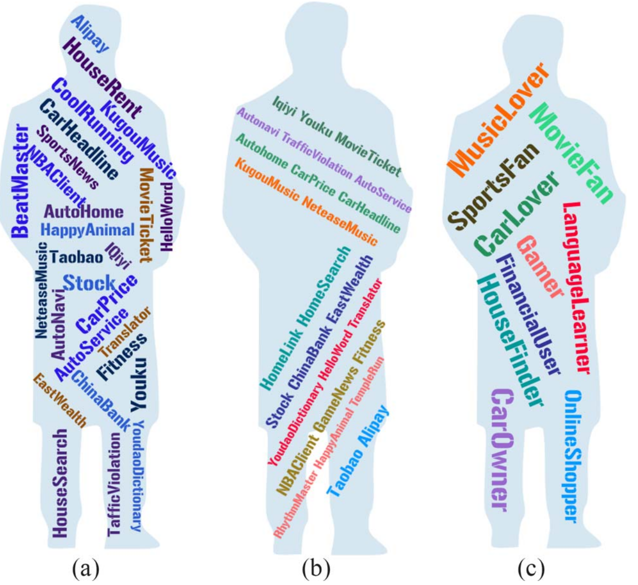


Fig. 11. Labeling user A with (a) apps, (b) basic component names, and (c) semantics of basic components.

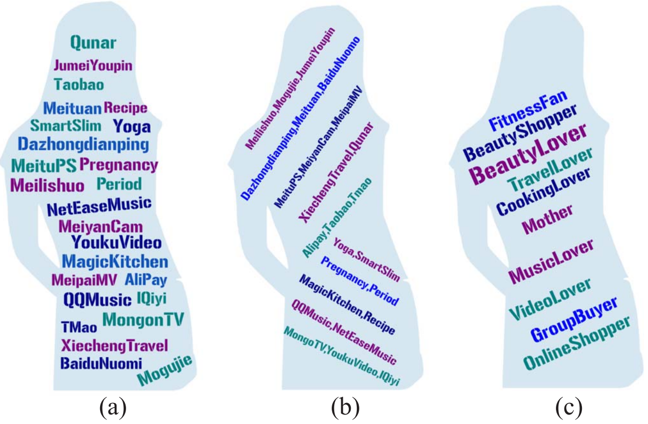


Fig. 12. Labeling user B with (a) apps, (b) basic component names, (c) semantics of basic components.

can be seen that for both of them each tag was very long and difficult to quickly capture the meaning.

VI. Conclusion

One's installed app list on a smartphone reveals lots of underlying user characteristics. In this article, we understood users through their installed app lists by building a compact and semantic feature space. A user representation framework was proposed, where we modeled the underlying relations between users and apps with the method of BMF. We discovered basic components of interrelated apps to build a compact user subspace. Basic components consisting of the interrelated apps reflect user needs and interests, which were used to represent each user. With user representation, we developed three typical application scenarios to understand users. More specifically, we mined user attributes of gender, smartphone price, and screen size from installed app lists by classification, achieving the accuracy of 77%, 93%, and 90%, respectively. We also discovered user groups using clustering, compared with other user subspaces. With the user representation framework, we showed how we can label each user with semantics extracted from the meaningful basic components.

Although installed app lists have lots of information about user attributes, they still have some defects. For example, installed app lists do not contain information on how frequently an app is used. It would be more informative if we know when and how much time an app is used. Besides, it is difficult to evaluate the semantic label given to each user. We will try to cope with these issues in the future work.

References

[1] B. H. Russell and P. D. Killworth, "Informant accuracy in social network data II," *Human Commun. Res.*, vol. 4, no. 1, pp. 3–18, 1977.

[2] I. Deutscher, *What We Say or What We Do: Sentiments and Acts*. Glenview, IL, USA: Scott Foresman, 1973.

[3] S. Seneviratne, A. Seneviratne, P. Mohapatra, and A. Mahanti, "Your installed apps reveal your gender and more!" *ACM SIGMOBILE Mobile Comput. Commun. Rev.*, vol. 18, no. 3, pp. 55–61, 2015.

[4] S. Zhao *et al.*, "Mining user attributes using large-scale app lists of smartphones," *IEEE Syst. J.*, vol. 11, no. 1, pp. 315–323, Mar. 2017.

[5] Z. Qin, Y. Wang, H. Cheng, Y. Zhou, Z. Sheng, and V. C. M. Leung, "Demographic information prediction: A portrait of smartphone application users," *IEEE Trans. Emerg. Topics Comput.*, vol. 6, no. 3, pp. 432–444, Jul.–Sep. 2018.

[6] G. Chittaranjan, J. Blom, and D. Gatica-Perez, "Mining large-scale smartphone data for personality studies," *Pers. Ubiquitous Comput.*, vol. 17, no. 3, pp. 433–450, 2013.

[7] R. M. Frey, R. Xu, and A. Ilic, "Mobile app adoption in different life stages: An empirical analysis," *Pervasive Mobile Comput.*, vol. 40, pp. 512–527, Sep. 2017.

[8] X. Zou, W. Zhang, S. Li, and G. Pan, "Prophet: What app you wish to use next," in *Adjunct Proc. 2013 ACM Conf. Pervasive Ubiquitous Comput.*, 2013, pp. 167-170.

[9] L. A. Fast and D. C. Funder, "Personality as manifest in word use: Correlations with self-report, acquaintance report, and behavior," *J. Pers. Soc. Psychol.*, vol. 94, no. 2, p. 334, 2008.

[10] F. Mairesse, M. A. Walker, M. R. Mehl, and R. K. Moore, "Using linguistic cues for the automatic recognition of personality in conversation and text," *J. Artif. Intell. Res.*, vol. 30, no. 1, pp. 457–500, 2007.

[11] S. C. Herring and J. C. Paolillo, "Gender and genre variation in weblogs," *J. Sociolinguistics*, vol. 10, no. 4, pp. 439–459, 2006.

[12] K. De Bock and D. Van den Poel, "Predicting website audience demographics forWeb advertising targeting using multi-website clickstream data," *Fundamenta Informaticae*, vol. 98, no. 1, pp. 49–70, 2010.

[13] Z. Yu, F. Yi, Q. Lv, and B. Guo, "Identifying on-site users for social events: Mobility, content, and social relationship," *IEEE Trans. Mobile Comput.*, vol. 17, no. 9, pp. 2055–2068, Sep. 2018.

[14] R. Xu, R. M. Frey, and A. Ilic, "Individual differences and mobile service adoption: An empirical analysis," in *Proc. IEEE BigDataService*, 2016, pp. 234–243.

[15] E. Malmi and I. Weber, "You are what apps you use: Demographic prediction based on user's apps," 2016. [Online]. Available: arXiv:1603.00059.

[16] Y. Wang, Y. Tang, J. Ma, and Z. Qin, "Gender prediction based on data streams of smartphone applications," in *Proc. BigCom*, 2015, pp. 115–125.

[17] S. Zhao *et al.*, "User profiling from their use of smartphone applications: A survey," *Pervasive Mobile Comput.*, vol. 59, Oct. 2019, Art. no. 101052.

[18] S. Zhao *et al.*, "Discovering different kinds of smartphone users through their application usage behaviors," in *Proc. ACM UbiComp*, 2016, pp. 498–509.

[19] H. Li *et al.*, "Characterizing smartphone usage patterns from millions of android users," in *Proc. ACM IMC*, 2015, pp. 459–472.

[20] Y.-X. Wang and Y.-J. Zhang, "Nonnegative matrix factorization: A comprehensive review," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 6, pp. 1336–1353, Jun. 2013.

[21] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *IEEE Comput.*, vol. 42, no. 8, pp. 30–37, Aug. 2009.

[22] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2001, pp. 556–562.

[23] P. Miettinen, T. Mielikainen, A. Gionis, G. Das, and H. Mannila, "The discrete basis problem," *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 10, pp. 1348–1362, Mar. 2008.

[24] R. Agrawal, T. Imielinski, and A. N. Swami, "Mining association rules between sets of items in large databases," in *Proc. ACM SIGMOD Int. Conf. Manag. Data*, 1993, pp. 207–216.

[25] A. Brankovic, A. Falsone, M. Prandini, and L. Piroddi, "A feature selection and classification algorithm based on randomized extraction of model populations," *IEEE Trans. Cybern.*, vol. 48, no. 4, pp. 1151–1162, Apr. 2018.

[26] A. Jain, M. Murty, and P. J. Flynn, "Data clustering: A review," *ACM Comput. Surveys*, vol. 31, no. 3, pp. 264–323, 1999.

[27] Y. Shen, W. Pedrycz, and X. Wang, "Clustering homogeneous granular data: Formation and evaluation," *IEEE Trans. Cybern.*, vol. 49, no. 4, pp. 1391–1402, Apr. 2019.

[28] A. Li, Z. Lu, L. Wang, P. Han, and J.-R. Wen, "Large-scale sparse learning from noisy tags for semantic segmentation," *IEEE Trans. Cybern.*, vol. 48, no. 1, pp. 253–263, Jan. 2018.

[29] E. Estellés-Arolas and F. González-Ladrón-De-Guevara, "Towards an integrated crowdsourcing definition," *J. Inf. Sci.*, vol. 38, no. 2, pp. 189–200, 2012.

[30] A. K. Jain, "Data clustering: 50 years beyond *k*-means," *Pattern Recognit. Lett.*, vol. 31, no. 8, pp. 651–666, 2010.

[31] T. M. Mitchell, *Machine Learning*. London, U.K.: McGraw-Hill, 1997.

Sha Zhao received the Ph.D. degree from Zhejiang University, Hangzhou, China, in 2017.

She is currently a Postdoctoral Research Fellow with the College of Computer Science and Technology, Zhejiang University. She visited the Human-Computer Interaction Institute, Carnegie Mellon University, Pittsburgh, PA, USA, as a visiting Ph.D. student from 2015 to 2016. Her research interests include pervasive computing, data mining, and machine learning.

Dr. Zhao received the Best Paper Award of ACM UbiComp'16.

Gang Pan (Member, IEEE) received the B.Eng. and Ph.D. degrees from Zhejiang University, Hangzhou, China, in 1998 and 2004, respectively.

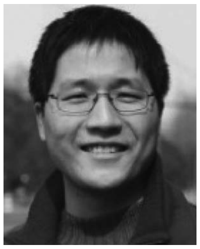
He is currently a Professor with the Department of Computer Science, and the Deputy Director of the State Key Lab of CAD&CG, Zhejiang University. From 2007 to 2008, he was a Visiting Scholar with the University of California, Los Angeles, CA, USA. His current interests include artificial intelligence, pervasive computing, brain-inspired computing, and brain-machine interfaces.

Jianrong Tao received the B.Sc. degree in computer science from the Huazhong University of Science and Technology, Wuhan, China, in 2014, and the master's degree in computer science from Zhejiang University, Hangzhou, China, in 2017.

He is currently a Algorithm Expert with Fuxi AI Lab, NetEase, Hangzhou. His research interests include machine learning and data mining.

Zhiling Luo received the B.S. and Ph.D. degrees in computer science from Zhejiang University, Hangzhou, China, in 2012 and 2017, respectively.

He was an Assistant Professor of computer science with Zhejiang University. He was a Visiting Scholar with the Georgia Institute of Technology, Atlanta, GA, USA, in 2016. His research interests include service computing, machine learning, and data mining.



Shijian Li received the Ph.D. degree from Zhejiang University, Hangzhou, China, in 2006.

In 2010, he was a Visiting Scholar with the Institute Telecom SudParis, Évry, France. He is currently with the College of Computer Science and Technology, Zhejiang University. He has published over 40 papers. His research interests include sensor networks, ubiquitous computing, and social computing.

Dr. Li serves as an Editor for the *International Journal of Distributed Sensor Networks* and as a reviewer or the PC member of more than ten conferences.

Zhaohui Wu (Fellow, IEEE) received the Ph.D. degree in computer science from Zhejiang University, Hangzhou, China, in 1993.

From 1991 to 1993, he was with the German Research Center for Artificial Intelligence as a joint Ph.D. student. He is currently a Professor of computer science with Zhejiang University, where he is the Director of the Institute of Computer System and Architecture. His current research interests include intelligent systems, semantic grid, and ubiquitous embedded systems.