From Recommendation to Profile Inference (Rec2PI): A Value-added Service to Wi-Fi Data Mining

Cheng Chen
University of Victoria, Canada
cchenv@uvic.ca

Fang Dong
University of Victoria, Canada
fdong@uvic.ca

Kui Wu
University of Victoria, Canada
wkui@uvic.ca

Venkatesh Srinivasan
University of Victoria, Canada
srinivas@uvic.ca

Alex Thomo
University of Victoria, Canada
thomo@uvic.ca

ABSTRACT

Portable smart devices have become prevalent and are used for ubiquitous access to the Internet in our daily life. Taking advantage of this trend, brick-and-mortar retailers have been increasingly deploying free Wi-Fi hotspots to provide easy Internet access for their customers. This opens the opportunity for retailers to collect customer information and perform data mining to improve the quality of their service. In this paper, we propose a novel value-added service to Wi-Fi data mining, Rec2PI, which can infer users' preference profiles based on recommendations pushed by third-party apps. Such profiles can be used to improve users' online experience and enable a brick-and-mortar retailer to participate in the global advertising business. Since the goal and technical difficulties of Rec2PI significantly differ from those of traditional recommender systems, we present a general framework of Rec2PI to illustrate its process. To tackle the technical challenges in profile inference, we propose novel algorithms built using copulas, a statistical tool suitable for capturing complex dependence structure beyond the scope of linear dependence. In the context of rating-based recommendations, we evaluate the proposed algorithms using an open dataset and a real-world recommender system. The evaluation results show that Rec2PI creates consistent and accurate inference results.

Keywords

Reverse Engineering of Recommendations; Wi-Fi Data Mining; Profile Inference; Copula Modelling

1. INTRODUCTION

In the current era of mobile technology, smart devices (e.g., smartphones, tablets, and smart wearables) have become more prevalent than ever before. Smart devices have provided people with ubiquitous access to the Internet, leading to an ever-growing ecosystem of Mobile Internet. Recent mobile marketing statistics shows that mobile users have outnumbered the desktop users worldwide and over 80% of mobile users access the Internet via smartphones. Following the trend of Mobile Internet, retailers with physical stores (the so-called brick-and-mortar retailers) are building their own wireless access points for smart devices to improve the user experience. Currently, free Wi-Fi services are offered in many places, including cafes, airports, hotels, restaurants, cinemas, and shopping malls.

Considering that retailers were reluctant to invest in Wi-Fi not so long ago, it is surprising to see that retailers are now embracing the in-store Wi-Fi. This opens up the opportunity that, in addition to improving customer satisfaction, the retailers could actually obtain a "goldmine" of customer data. With the free Wi-Fi services provided, retailers can collect useful information about their customers such as their geographic data and dwell times at different locations. The data, which the customers opt to share, offers retailers a better understanding of customers' behavior and demographics and helps them make informed marketing decisions.

Currently, analytics based on Wi-Fi data has become the focus of many Wi-Fi provider companies, such as AirTight Wi-Fi[1] and Purple Wi-Fi[2]. They help retailers not only deploy Wi-Fi, but also launch analytics engines for in-store business intelligence and customer engagement. This rapidly evolving market has also attracted the attention of the government agencies. Recently, New York is transforming old phone booths into city-wide free Wi-Fi hotspots, that can collect pertinent information for the purpose of targeted advertising.

Compared to major Ecommerce companies such as Amazon and eBay that have a great amount of data to learn user behavior, a brick-and-mortar retailer generally can only collect a limited amount of data. The problem of providing competitive value-added services based on Wi-Fi collected data remains open and has not been well studied in the literature. Most existing industry solutions mine the collected data for basic customer demographics, presence analytics, Wi-Fi usage, and loyalty and engagement. A natural question is: can we gain more knowledge on the customer preference profile for products of interest using a very limited amount of data collected by the Wi-Fi service provider?

In fact, abundant information is hidden in the small amount

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

CIKM'16 , October 24-28, 2016, Indianapolis, IN, USA

© 2016 ACM. ISBN 978-1-4503-4073-1/16/10…$15.00

DOI: http://dx.doi.org/10.1145/2983323.2983827

[1]http://www.mojonetworks.com/, Accessed Jan. 2016
[2]http://purple.ai/enterprise/, Accessed Jan. 2016

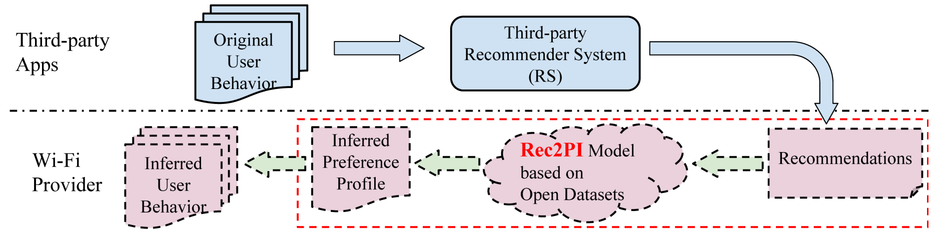


Figure 1: The general framework of Rec2PI model (red dashed box) and the work flow.

of Wi-Fi data. The mobile apps offer the customers a wide range of services such as social networking, shopping, and entertainment. After connecting to the in-store Wi-Fi, apps with integrated recommender systems normally push recommendations to the user based on the user's past behavior or preference (e.g., purchase history, ratings) and current environment. For example, a customer who is visiting a book store may receive recommended books that the customer might purchase. By exploring the hidden knowledge behind the recommendations, the Wi-Fi provider can practically learn more information regarding this customer.

Recommendation data can be obtained by tracking the user's browsing behavior. In the service agreement (or the privacy policy) of free Wi-Fi services, such as RetailNext, CityBridge and Target, the providers are allowed to collect users' browsing behavior (e.g., URLs, pages visited, etc.) if the users choose to use the free service. In this work, we assume that the recommendations pushed by the mobile-apps are (partially) available to the free Wi-Fi service provider, who aims to infer the customer's preference profile based on the collected recommendations. While service agreements of real-world free Wi-Fi providers allow them to collect the above information, the debate on privacy and ethical concerns is out of the scope of the paper. We believe that being able to infer the preference profile of a customer is beneficial for: (a) the customer because he will be offered a better range of products, enabling him to obtain the best product cheaper, and (b) the brick-and-mortar retailer by enabling it to participate in the global advertising business and increase its revenues.

In this paper, we introduce and study a novel value-added service, Recommendation to Profile Inference (Rec2PI), for Wi-Fi data mining. Rec2PI utilizes a new source of data, i.e., recommendations pushed to a user in a certain domain (e.g., books or movies), to infer the user's preference profile in that domain. Figure 1 depicts a general framework, where the red dashed box contains the input, the inference model, and the output of Rec2PI. When a target user logs into the in-store Wi-Fi for Internet access, third-party mobile apps may push recommendations to the user. After collecting the pushed recommendations, Rec2PI infers the user's preference. Once the inferred preference profile is obtained, the Wi-Fi provider can further estimate the customer's behavior, which is valuable for retailers to deliver more personalized services.

Rec2PI is significantly different from any existing work of recommender systems (RS). Traditional RS has universal access to a user's ground truth (e.g., past purchase behavior and item ratings). On the contrary, Rec2PI may not possess this information. As such, Rec2PI can be viewed as a reverse of the learning procedure in RS. The main challenge behind Rec2PI is: without knowing the algorithm(s) and the dataset behind the third-party RS, how can we effectively infer the user's preference profile?

This paper answers the above challenge. To make the reverse learning possible, we make use of open datasets in the same domain as the recommendations. For example, Epinions is well-known for customer reviews about products. IMDB and MovieLens are popular sources for movie average ratings, meta information, and individual ratings. With the knowledge from the open datasets, we can learn the most likely user factors that would result in the recommendations. With this intuition, we make the following contributions in the paper:

- We initiate the study of a novel value-added service arising in Wi-Fi data mining: Without knowing the algorithms and the dataset used by a third-party RS, how can we infer users' behavior based on the recommendations from the third-party RS? To the best of our knowledge, we are the first to investigate this reversed learning problem.

- We propose a general framework, Rec2PI, that builds probabilistic inference models based on open datasets. In addition, we adopt a novel approach that incorporates copulas, a powerful statistical tool for dependence modeling, into the inference procedure.

- We perform extensive experimental evaluation on real-world datasets. We show that the performance of popular approaches in RS, such as latent factor models (LFMs), is not stable when solving the reversed learning problem, i.e., the results exhibit high variance. In contrast, our copula-based solution is not only accurate but also much more stable.

2. ASSUMPTIONS AND PRELIMINARIES

2.1 Assumptions

To introduce Rec2PI, we begin with a high-level model shown within the red dashed box in Figure 1. It consists of the recommendations from third-party RS (the input), the inference model, and inferred user preference profile (the output). Based on the inferred user profile, Rec2PI further estimates the user behavior. Our goal is to make the inferred user behavior as close as possible to the original.

Following the notations defined in Table 1, we make the following assumptions.

- Let $\mathcal{F} : r_t \rightarrow \hat{\mathcal{R}}_t$ denote the recommendation model of the third-party app's RS. We will assume that $\mathcal{F}$, as well as the dataset it uses, are hidden from the Wi-Fi provider.

Table 1: Notations of General Rec2PI

Notation	Explanation	Accessible to Wi-Fi Provider?
t	a target user t	Yes
r_t	t's original behavior	No
r'_t	t's inferred behavior	Yes
$\mathcal{F}$	recommendation model of the third-party RS	No
Λ_t	t's inferred preference profile	Yes
$\hat{\mathcal{R}}_t$	recommendations by third-party RS to t	Yes
R	the open dataset	Yes
U	the user set of the open dataset	Yes
I	the item set of the open dataset	Yes
M_r	the highest rate value	Yes

- The Wi-Fi provider has the access to an open dataset (called open dataset hereafter), which belongs to the same category as the dataset that the third-party RS uses (called hidden dataset hereafter). For example, both are movie datasets or both are book datasets. Nevertheless, there is no guarantee that the two datasets are identical.

- Rec2PI does not rely on any hypothesis regarding the recommendation algorithm used by $\mathcal{F}$.

In Rec2PI, we first need to determine a method to represent users' preference profile, Λ_t. There are different ways to represent preference profiles, including the vector representation [5], ontology representation [15], and multidimensional representation [12]. Among these, the latent factor model (LFM) [11], a well-known variant of the vector representation, has been the most popular one. We adopt LFM in the paper for users' preference profile. We stress that LFM is only used in Rec2PI. The third-party RS does not necessarily use LFM inside $\mathcal{F}$.

2.2 Background of Latent Factor Models

Latent factor models (LFMs) assume that a user's behavior (e.g., purchases and ratings) is influenced by a set of latent factors. The term "latent" implies that these factors do not necessarily correspond to physical meanings. LFMs serve as one of the most popular collaborative filtering (CF) techniques in rating-based item recommendation [11].

Definition 1 (User Behavior). *The behavior of a user u is captured by a vector of rating, denoted by $\mathbf{r}_u$ as follows:*

$$\mathbf{r}_u = [r_{u,i_1}, r_{u,i_2}, \ldots, r_{u,i_n}]^T, \tag{1}$$

where $i_j (j = 1, \ldots, n)$ denote the items that user u has rated and r_{u,i_j} denotes the rating that the user gave on item i_j.

Definition 2 (User Preference Profile). *The (latent) preference profile of user u, denoted as Λ_u, is a D-dimensional vector,*

$$\Lambda_u = [u^{(1)}, u^{(2)}, \ldots, u^{(D)}]^T, \tag{2}$$

where each $u^{(d)}, d = 1, \ldots, D$, is called a latent factor of user u, and D is the total number of latent factors.

Definition 3 (Item Latent Profile). *The (latent) profile of item i, denoted as Γ_i, is also a D-dimensional vector,*

$$\Gamma_i = [i^{(1)}, i^{(2)}, \ldots, i^{(D)}]^T, \tag{3}$$

where each $i^{(d)}, d = 1, \ldots, D$, is called a latent factor of item i, and D is the total number of latent factors.

The latent profile Λ_u represents the user's preference in the D-dimensional latent factor space, and Γ_i captures item i's feature in the D-dimensional latent factor space. A LFM produces predicted rating that user u gives to i, $r'_{u,i}$, as

$$r'_{u,i} = \Lambda_u^T \cdot \Gamma_i. \tag{4}$$

A LFM thus tries to learn the latent factors for all users and all items, by minimizing the regularized squared error [11]:

$$\underset{\Lambda_u, \Gamma_i, \theta_1, \theta_2}{\operatorname{argmin}} \sum_{r_{u,i} \in \mathbf{R}} (r_{u,i} - \Lambda_u^T \cdot \Gamma_i)^2 + \theta_1 \cdot ||\Lambda_u||^2 + \theta_2 \cdot ||\Gamma_i||^2, \tag{5}$$

where $\mathbf{R} = (r_{u,i})_{|U| \times |I|}$ represents a (sparse) rating matrix which contains the ground truth ratings for $u \in U$ and $i \in I$, U and I denote the set of users and the set of items, respectively. This can be seen as factorizing a rating matrix $\mathbf{R}$ into a user factor matrix and an item factor matrix.

3. PROBLEM FORMULATION

3.1 The Goal of Rec2PI

Definition 4 (User's Recommendations). *The recommendations from the third-party RS to a target user t, $\hat{\mathcal{R}}_t$, is a vector of ratings[3] as follows:*

$$\hat{\mathcal{R}}_t = [\hat{r}_{t,i_1}, \hat{r}_{t,i_2}, \ldots, \hat{r}_{t,i_{\hat{n}}}]^T, \tag{6}$$

where $i_j \in \hat{I}_t$ is an item in t's recommended item set $\hat{I}_t$ with rating $\hat{r}_{t,i_j}$, and $\hat{n} = |\hat{I}_t|$.

Denote $\mathcal{G} : \hat{\mathcal{R}}_t \to \Lambda_t$ as the inference function of Rec2PI. **The goal of Rec2PI is thus to estimate Λ_t by applying $\mathcal{G}$ into an open dataset.** Once Λ_t is available, Rec2PI can use LFM to produce the inferred user behavior r'_t.

3.2 Why Does Traditional RS Not Work for Rec2PI?

Estimating Λ_t based on $\hat{\mathcal{R}}_t$ is fundamentally different from traditional RS, due to the fact that (1) the open dataset is not the same as the hidden dataset and the target user may not exist in the open dataset, (2) we may not have any

[3] A prominent RS that sends predicted ratings along with recommended items is Netflix (refer to Section 5.1 as well).

ground truth rating[4] from the target user. Consequently, Rec2PI needs to tackle two challenges:

1. Recommended item ratings $\hat{\mathcal{R}}_t$ cannot be used in the same way as the ground truth ratings in traditional RS that uses the ground truth ratings to obtain latent factors by solving (5). Recommended item ratings are determined by a specific third-party RS and do not necessarily correspond to the user's true ratings.

2. The details of $\mathcal{F}$ is hidden from the Wi-Fi provider, meaning that Rec2PI has no knowledge on how recommended ratings are generated. Since there are many recommendation algorithms even in the same domain (e.g., movie recommendation), we cannot assume a specific algorithm for $\mathcal{F}$. Guessing the algorithms behind a *proprietary* RS is still an open challenge.

3.3 Intuition and Discussion

One may wonder about two obstacles we need to overcome in Rec2PI: why is it possible to infer a user's profile based on the recommendations generated with some unknown algorithm over a hidden dataset? how can the accuracy of the inference results be evaluated without knowing the original user behavior (ground truth)?

To answer the first question, we give an intuitive explanation before we dive into technical details. In principle, an RS makes recommendations based on the *dependence* between the features of items and the flavor of the user. This dependence structure reflects the statistical patterns *inherent* in the real-world phenomenon, e.g., males tend to love action movies more than females. These statistical patterns can be found in different datasets. In other words, it is the statistical patterns instead of the uniqueness of dataset that determine the recommendation results. Two different datasets in the same domain (e.g., user-movie ratings), even if they are not identical, should exhibit similar statistical patterns as long as both are large enough to reflect the real-world phenomenon. This explains why we can infer a user's profile based on the recommendations generated from a hidden dataset using an unknown recommendation algorithm.[5]

To answer the second question, we design a special evaluation method to avoid the ground-truth problem in the evaluation of Rec2PI. Our idea is to randomly select users from the open dataset so that we know their ground-truth behavior. For each selected user, we manually create an "agent" user in the third-party service and manually set the ratings of the "agent" user the same as the ones in the corresponding user in the open dataset. Note that since the third-party datasets and open datasets belong to the same category (e.g., books or movies), it is reasonable to assume that items in both datasets have overlaps and that we can find corresponding items in the third-party datasets for rating assignments. The recommendations to the "agent" user by the third-party RS will be used as the input to Rec2PI. The inferred behavior by Rec2PI for the "agent" user is compared to the ground-truth behavior in the corresponding user in the open dataset. More details will be disclosed in Section 5.

3.4 A New Approach

To overcome the difficulties raised in Section 3.2, we model Rec2PI in a new approach different from traditional RS. Specifically, we only assume limited prior knowledge about $\mathcal{F}$, i.e., $\mathcal{F}$ is trained on a certain type of hidden datasets (e.g., user-item ratings) in a particular category. Although Rec2PI does not have access to the hidden datasets, it can make use of similar types of open datasets in the same category and learn similar patterns using common methods. The open datasets do not necessarily contain the target user's records, and therefore we do not attempt to infer t's profile with Equation (5). Instead, we consider each recommended item $i \in \hat{I}_t$ individually and explore its association with the user, i.e., what type of users are likely to be interested in the recommended item. In other words, given a recommended item, we first obtain its latent factors using the open dataset and then calculate the most-likely latent factors that a user should have such that the item would be recommended to the user.

This task requires us to study the dependence between user latent factors and item latent factors. To model the relationship in a probabilistic approach, we first denote $\mathcal{U} = \{\mathcal{U}^{(1)}, \dots, \mathcal{U}^{(D)}\}$ as a set of continuous random variables for each dimension of user factors, and $\mathcal{I} = \{\mathcal{I}^{(1)}, \dots, \mathcal{I}^{(D)}\}$ as a set of continuous random variables for each dimension of item factors. Let f denote the probability density function (PDF) and F the cumulative density function (CDF). In addition, let $\mathbf{E}$ denote the set of observed evidence variables (i.e., item factors and the ratings in the open dataset). Given a set of recommended items $\hat{I}_t$ with ratings $\hat{\mathcal{R}}_t$, Rec2PI solves a series of the maximum a posteriori probability estimation problems (MAP) as follows:

$$\mathrm{MAP}(\mathcal{U}|\mathbf{E}) = \operatorname*{argmax}_{\mathbf{u}} f(\mathbf{u}|\mathbf{i}, \hat{r}_{t,i}), \forall i \in \hat{I}_t, \tag{7}$$

where $\mathbf{u}$ and $\mathbf{i}$ represent values for random variables in $\mathcal{U}$ and $\mathcal{I}$, respectively.

4. COPULA-BASED PROBABILISTIC PROFILE INFERENCE

4.1 Outline of Solution

We introduce our proposed method to solve the problem formulated in Equation (7) by converting the equation to a more detailed form in our context. Without loss of generality, we assume that $\hat{r}_{t,i}$ has an integer range[6]. Denote the rating matrix in the open dataset as $\mathbf{R}$. Denote the range of ratings as $[1, M_r]$. It is reasonable to assume that the item sets (e.g., movies) in the hidden dataset (used in the third-party RS) and the open dataset (explored by Rec2PI) are close, because we have the freedom to use any open dataset that sufficiently covers all pushed items to the target user.

Ratings of recommended items, predicted by the (unknown) recommendation algorithm $\mathcal{F}$, reflect the statistical patterns in user-item association. Since we do not attempt to minimize the distance metric such as that defined in Equation (5), we solve the problem based on the fact that the ratings reflect the dependence structure between item factors

[4] Ground truth ratings of the user in consideration are part of the input in traditional RS, because RS will not be able to create a personalized recommendation if the user has not rated any item.

[5] The unknown algorithm is assumed to make reasonable recommendations that reflect the statistical patterns.

[6] Decimal ratings can be converted into integers.

and user factors. As such, for each rating value $x \in [1, M_r]$, we associate x with $\mathcal{U}_x = \{\mathcal{U}_x^{(1)}, \dots, \mathcal{U}_x^{(D)}\}$, the set of continuous random variables for each dimension of user factors, and with $\mathcal{I}_x = \{\mathcal{I}_x^{(1)}, \dots, \mathcal{I}_x^{(D)}\}$, the set of continuous random variables for each dimension of item factors. The details on how to obtain sample values of $\mathcal{U}_x$ and $\mathcal{I}_x$ from $\mathbf{R}$ will be given in Subsection 4.3.

A common assumption in LFM is that the latent user factors are independent of each other [16, 14]. This assumption suggests that we can infer the D-dimensional user factors one at a time. Therefore, corresponding to a recommended item $i \in \hat{I}_t$ with rating $\hat{r}_{t,i}$, we can infer user t's d-th latent factor, conditioning on i, denoted as $\Lambda_{t,i}^{(d)}$, as

$$\Lambda_{t,i}^{(d)} = \underset{u_x^{(d)}}{\operatorname{argmax}}\, f(u_x^{(d)}|\mathbf{i}_x), \forall i \in \hat{I}_t, \tag{8}$$

where to simplify notation $x = \hat{r}_{t,i}$ and $\mathbf{i}_x = \{i_x^{(1)}, \dots, i_x^{(D)}\}$. Note that $\mathbf{i}_x = \{i_x^{(1)}, \dots, i_x^{(D)}\}$ represent the values for random variables in $\mathcal{I}_x = \{\mathcal{I}_x^{(1)}, \dots, \mathcal{I}_x^{(D)}\}$.

We then take the average value over all $\Lambda_{t,i}^{(d)}, \forall i \in \hat{I}_t$ as the final value of inferred d-th latent factor, i.e.,

$$\Lambda_t^{(d)} = \frac{1}{|\hat{I}_t|} \sum_{i \in \hat{I}_t} \Lambda_{t,i}^{(d)}. \tag{9}$$

Nevertheless, $f(u_x^{(d)}|\mathbf{i}_x)$ in Equation (8) is non-trivial to model because it involves the dependence structure between user factors and item factors. In the next subsection, we will introduce a powerful statistical method, copula modeling, that can characterize this dependence information.

4.2 Why Copula-based Inference?

We start the introduction to copulas with the definition of 2-dimensional (bivariate) copulas and core theorems.

Definition 5. *A 2-dimensional copula is a function C having the following properties [18]:*

1. *Its domain is $[0, 1]^2$;*
2. *C is 2-increasing, i.e., for every $u_1, u_2, v_1, v_2 \in [0, 1]$ and $u_1 \le u_2, v_1 \le v_2$, we have $C(u_2, v_2) - C(u_2, v_1) - C(u_1, v_2) + C(u_1, v_1) \ge 0$.*
3. *$C(u, 0) = C(0, v) = 0$, $C(u, 1) = u$, $C(1, v) = v$, for every $u, v \in [0, 1]$.*

Theorem 1. ***(Sklar's theorem)** [18] Consider two random variables X and Y, with $F(x, y)$ as the joint CDF. Denote the marginal CDF of X and the marginal CDF of Y as $F_X(x)$ and $F_Y(y)$, respectively. Then there exists a copula C such that for all x and y, $F(x, y) = Pr(X \le x, Y \le y) = C(F_X(x), F_Y(y))$.*

Sklar's theorem shows how the copula models the dependence between univariates. The copula is a function that links univariate marginals to their joint distributions. This property is especially useful since the joint distribution of random variables is difficult to find directly in many applications. For example, in our problem, when considering a feature of users and a feature of items as two random variables, their joint distribution is not easy to obtain. By using copulas to model the dependence, their joint behavior can be revealed based on Theorem 1 by integrating marginal distribution into a copula model.

Copulas are efficient and well known for dependence modeling and have been widely used in finance and risk management [7, 18]. There are a variety of copula families which are well studied, such as Archimedean copulas, Gaussian copula, student's t copula, and etc. These known copulas are powerful tools to capture different types of dependence structure [7], so that we will have enough choices when using copulas for dependence modelling in our problems. In addition, compared with other dependence measurements, such as covariance and correlation, copulas not only capture the linear dependence, but also model the functional dependence between random variables [7]. In the context of Rec2PI, there is no evidence that the dependence structure between users and items is linear. In this case, copulas can be used to model this dependence well even if the dependence goes beyond the linear scope.

The following theorem is also useful in our copula-based model.

Theorem 2. *[2] Assume X and Y have the copula C between them, the conditional density function $f(Y = y|X = x)$ is*

$$f(Y = y|X = x) = c(F_X(x), F_Y(y))f_Y(y) \tag{10}$$

where $c(u, v) = \frac{\partial}{\partial u}\frac{\partial}{\partial v}C(u, v)$ is called the copula density function, and $f_Y(y)$ is the probability density function (PDF) of the marginal.

4.3 Copula-based Probabilistic Model (CPM)

To ease notation, in the rest of the paper, we use f to denote the PDF for univariate, bivariate and multivariate and F to denote the CDF for univariate, bivariate and multivariate, without using subscripts. For instance, $F(i_x^{(1)})F(i_x^{(2)}) \cdots F(i_x^{(D)})$ should be read as $F_{\mathcal{I}_x^{(1)}}(i_x^{(1)})F_{\mathcal{I}_x^{(2)}}(i_x^{(2)}) \cdots F_{\mathcal{I}_x^{(D)}}(i_x^{(D)})$.

We use the copula method to solve Equation (8). To simplify calculation, we in this section assume that the item factors are independent of each other. In the next section, we will relax this assumption and introduce an improved algorithm that can handle the dependence between item factors.

Assume $\hat{r}_{t,i} = x$. Due to the independence assumption on item latent factors, $F(\mathbf{i}_x)$ can be computed as:

$$\begin{aligned} F(\mathbf{i}_x) &= F(i_x^{(1)}, i_x^{(2)}, \dots, i_x^{(D)}) \\ &= F(i_x^{(1)})F(i_x^{(2)}) \cdots F(i_x^{(D)}) \end{aligned} \tag{11}$$

Based on Theorem 2, Equation (8) can be rewritten as:

$$\begin{aligned} \Lambda_{t,i}^{(d)} &= \underset{u_x^{(d)}}{\operatorname{argmax}}\, f(u_x^{(d)}|\mathbf{i}_x) \\ &= \underset{u_x^{(d)}}{\operatorname{argmax}}\, c_x(F(u_x^{(d)}), F(\mathbf{i}_x))f(u_x^{(d)}), \quad \forall i \in \hat{I}_t, \end{aligned} \tag{12}$$

where c_x is the bivariate copula density function of $\mathcal{U}_x^{(d)}$ and $\mathcal{I}_x$. Note that we transform $\mathcal{I}_x$ as one random variable with Equation (11). From Equation (12), the inference on d-th latent factor will be made by modelling its dependence with recommended items and its own marginals.

Next, we present an algorithm that, for each rate value $x \in [1, M_r]$, constructs the sample values for $\mathcal{U}_x = \{\mathcal{U}_x^{(1)}, \dots, \mathcal{U}_x^{(D)}\}$ and $\mathcal{I}_x = \{\mathcal{I}_x^{(1)}, \dots, \mathcal{I}_x^{(D)}\}$, using the open dataset $\mathbf{R}$.

Let N_x denote the total times of occurrence for a rating value x in $\mathbf{R}$. For a rating value x, let a $(N_x \times D)$-dimensional matrix $\mathbf{L}_U^{(x)}$ denote the user-side matrix where each row is a vector of user latent factors, and let a $(N_x \times D)$-dimensional matrix $\mathbf{L}_I^{(x)}$ denote an item-side matrix where each row is a vector of item latent factors. Algorithm 1 presents the details of constructing $\mathbf{L}_U^{(x)}$ and $\mathbf{L}_I^{(x)}$, for all $x \in \{1, \ldots, M_r\}$.

Algorithm 1: ***Preprocessing for Latent Factors***

Input: A rating matrix $\mathbf{R}$, user set U and item set I
Output: $\mathbf{L}_U^{(x)}$ and $\mathbf{L}_I^{(x)}$, $\forall x \in \{1, \ldots, M_r\}$

1 Learn a LFM model by factorizing $\mathbf{R}$ with Equation (5) to obtain D-dimensional latent factors Λ_u for $u \in U$ and Γ_i for $i \in I$;
2 For all $r_{u,i} \in \mathbf{R}$, insert Λ_u^T into $\mathbf{L}_U^{(r_{u,i})}$, and insert Γ_i^T into $\mathbf{L}_I^{(r_{u,i})}$;

Algorithm 2: ***Bivariate-CPM***

Input: $\mathbf{L}_U^{(x)}$ and $\mathbf{L}_I^{(x)}$, $x \in \{1, \ldots, M_r\}$, and a set of recommended items $\hat{I}_t$ with ratings $\hat{\mathcal{R}}_t$
Output: The target user t's inferred profile $\Lambda_t = [t^{(1)}, t^{(2)}, \ldots, t^{(D)}]^T$

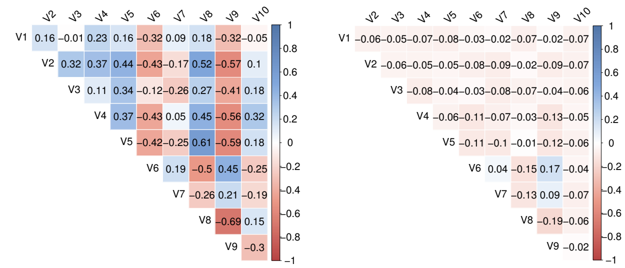
1 **foreach** x *in* $\{1, \ldots, M_r\}$ **do**
2   **foreach** d *in* $\{1, \ldots, D\}$ **do**
3   Fit a Gaussian distribution $\mathcal{N}(\mu_{\mathcal{U}_x^{(d)}}, \sigma_{\mathcal{U}_x^{(d)}})$ to the d-th column of $\mathbf{L}_U^{(x)}$, i.e., $\mathbf{L}_U^{(x)}[:, d]$;
4   Fit a Gaussian distribution $\mathcal{N}(\mu_{\mathcal{I}_x^{(d)}}, \sigma_{\mathcal{I}_x^{(d)}})$ to the d-th column of $\mathbf{L}_I^{(x)}$, i.e., $\mathbf{L}_I^{(x)}[:, d]$;
5   Set $\mathcal{U}_x^{(d)} \sim \mathcal{N}(\mu_{\mathcal{U}_x^{(d)}}, \sigma_{\mathcal{U}_x^{(d)}})$ and $\mathcal{I}_x^{(d)} \sim \mathcal{N}(\mu_{\mathcal{I}_x^{(d)}}, \sigma_{\mathcal{I}_x^{(d)}})$;
6   Compute a CDF value for each entry in the d-th column of $\mathbf{L}_U^{(x)}$ as $F(u_x^{(d)})$;
7   Compute a CDF value for each row of $\mathbf{L}_I^{(x)}$ as $F(\mathbf{i}_x)$ by Equation (11) ;
8   Fit a bivariate copula $c_x(F(u_x^{(d)}), F(\mathbf{i}_x))$ to the CDF values of $F(u_x^{(d)})$ and $F(\mathbf{i}_x)$;
9 **foreach** d *in* $\{1, \ldots, D\}$ **do**
10   $\Lambda_t^{(d)} = \frac{1}{|\hat{I}_t|} \sum_{i \in \hat{I}_t} \operatorname{argmax}_{u_x^{(d)}} c_x(F(u_x^{(d)}), F(\mathbf{i}_x)) f(u_x^{(d)})$;
11 **return** $\Lambda_t = [t^{(1)}, t^{(2)}, \ldots, t^{(D)}]^T$;

As an example, consider the case where any rating value $x \in \{1, 2, 3, 4, 5\}$. Algorithm 1 outputs 5 user-side matrices $\{\mathbf{L}_U^{(1)}, \cdots, \mathbf{L}_U^{(5)}\}$, and 5 item-side matrices $\{\mathbf{L}_I^{(1)}, \cdots, \mathbf{L}_I^{(5)}\}$.

In Algorithm 2, we describe the training and inference procedures. We place Gaussian priors over $\mathcal{U}_x^{(d)}$ and $\mathcal{I}_x^{(d)}$, i.e., $\mathcal{U}_x^{(d)} \sim \mathcal{N}(\mu_{\mathcal{U}_x^{(d)}}, \sigma_{\mathcal{U}_x^{(d)}})$ and $\mathcal{I}_x^{(d)} \sim \mathcal{N}(\mu_{\mathcal{I}_x^{(d)}}, \sigma_{\mathcal{I}_x^{(d)}})$. The samples for $\mathcal{U}_x^{(d)}$ is the d-th column of $\mathbf{L}_U^{(x)}$. Similarly, the samples for $\mathcal{I}_x^{(d)}$ is the d-th column of $\mathbf{L}_I^{(x)}$.

4.4 Vine-copula Probabilistic Model (VPM)

In **Line 7** of Algorithm 2, we assume that the D random variables of item latent factors are independent, i.e., $\mathcal{I}_x^{(j)} \perp\!\!\!\perp \mathcal{I}_x^{(k)}, j \neq k$, so that we can transform $\mathcal{I}_x$ as one random variable with Equation (11) and apply bi-variate copula modeling. In this section, we relax this assumption and propose a vine-copula probabilistic model (VPM) to explore the dependence between item factors for further improvement.



(a) Item factors (b) User factors

Figure 2: Correlation coefficients (whose values correspond to the right color bar) between 10-dimensional of latent factors (rating $x = 3$).

Figure 2 shows the correlation coefficient values (i.e., linear dependence[7]) among 10 user/item factors learned from LFM on a movie ratings dataset, the details of which will be described in Section 5. While the dependence between user factors is weak, we can easily observe the strong correlations between item factors (e.g., $V2$-$V9$, $V5$-$V8$). Therefore, by capturing the dependence structure among the D random variables for item factors, we may be able to obtain better inference results in Rec2PI.

While so far we introduced copulas for the bivariate case, all definitions and theorems in Subsection 4.2 can be extended to multivariate copulas [18]. In this section, multivariate copulas are used to capture the dependence between item factors, i.e., to calculate the joint CDF $F(\mathbf{i}_x) = F(i_x^{(1)}, i_x^{(2)}, \ldots, i_x^{(D)})$.

A general approach to construct multivariate copulas is to extend a bivariate copula into the high dimensional version. Such extension has been made for multivariate Gaussian copula and multivariate student's t copula. Taking multivariate Gaussian copula as an example, it essentially models the dependence between any pairs of the multivariates with a bivariate Gaussian copula. This approach, however, is inflexible in high dimensions. In addition, since it constrains all pairs of random variables with the same dependence structure, the modelling power is limited.

Another construction method for multivariate copulas is to decompose the multivariate distribution into products of marginal PDFs and bivariate copulas PDFs. This method is called pair-copula construction (PCC). With PCC, each pair-copula can be chosen independently from the others, making the model more flexible. For high dimensional distributions, PCC often results in a great number of possible pair-copula constructions. Brechmann et al. [3] proposed to organize these constructions using a graphical model involving a sequence of nested trees, which are denoted as *regular vines*. There are two popular special cases of regular vines: the canonical (C-) vine and the D-vine. Each vine decomposes a multivariate distribution in a specific structure. In

[7]Note that the linear dependence between item factors does not suggest/imply the linear dependence between users and items.

this work, we adopt D-vine in our algorithm. The D-vine [3] decomposes a D-dimensional multivariate PDF $f(\mathbf{x})$ as

$$f(\mathbf{x}) = \prod_{d=1}^{D} f(x_d) \times \prod_{i=1}^{D-1} \prod_{j=1}^{D-i} c_{j,j+i|j+1,\ldots,(j+i-1)}(F(x_j|x_{j+1},\ldots,x_{j+i-1}), F(x_{j+i}|x_{j+1},\ldots,x_{j+i-1})), \quad (13)$$

where $f(x_d), d = 1, \cdots, D$, denote the marginal PDFs and $c_{j,j+i|j+1,\ldots,(j+i-1)}$ denote bivariate copula densities. Joe [10] showed that marginal conditional distributions of the form $F(x|\mathbf{v})$ in Equation (13) can be computed as:

$$F(x|\mathbf{v}) = \frac{\partial C_{x|v_j}(F(x|\mathbf{v}_{-j}), F(v_j|\mathbf{v}_{-j}))}{\partial F(v_j|\mathbf{v}_{-j})} \quad \forall v_j \in \mathbf{v}, \quad (14)$$

where v_j is any element of $\mathbf{v}$ and $\mathbf{v}_{-j}$ denotes the vector without this element. Once $f(\mathbf{x})$ is calculated, we use sampling to estimate the joint CDF $F(\mathbf{x})$. Given any instance $\mathbf{i}$ (values of an item's factors), the idea is to generate S samples from $f(\mathbf{x})$, and count the number of samples s satisfying $x^{(d)} \leq i^{(d)}, \forall d \in \{1, \ldots, D\}$. Then $F(\mathbf{i}_x)$ in Line 7 of Algorithm 2 can be calculated as $F(\mathbf{i}_x) \approx s/S$.

To summarize, VPM is also based on Algorithm 2, except that it uses Equations (13), (14) and a sampling technique to calculate $F(\mathbf{i}_x)$ in Line 7 of Algorithm 2.

5. EXPERIMENTAL EVALUATION

We conduct experiments to evaluate our proposed methods, CPM and VPM, in the scenario of movie preference inference. To avoid the ground-truth problem raised in Section 3.3, we design an evaluation method shown in Figure 3, whose details will be given in the next subsection. Note that the experimental steps shown in Figure 3 is for purpose of evaluation only. The true work flow of Rec2PI in real world should follow Figure 1. In addition, we perform evaluation in the movie domain not only because it is an important source of preference information, but also because it allows us to conveniently collect item recommendations generated by third-party RS (details explained in the next subsection), which are the input to Rec2PI but are often not included in publicly available datasets. Rec2PI can be applied to data in other domains as well, such as transactional datasets (purchases as binary ratings) in Ecommerce.

5.1 Data Preparation and Evaluation Steps

In the experiments, we use MovieLens as the open dataset to train Rec2PI for the Wi-Fi provider side and use Netflix's recommender system to create recommendations for target users. We describe how we select target users and collect recommended movie ratings in the following steps:

1. We use the latest (released in April, 2015) MovieLens-20m dataset[8] as the open dataset. We first filter movies in MovieLens to retain those appearing in both MovieLens and Netflix.

2. We randomly select target users from MovieLens (so that ground truth ratings can be obtained) and exclude them as well as the associated ratings from the dataset used for evaluation afterwards.

3. For each target user, we randomly choose a set of titles of his/her unrated movies from MovieLens. Movies in this set serve as the recommended ones whose predicted ratings will be determined by Netflix's RS.

4. For each target user, we create an individual "agent" account on Netflix and manually assign the ground truth ratings to the corresponding movies for the account.

5. On each target user's account, we search the titles of the selected unrated movies on Netflix's RS. We then record Netflix's predicted ratings.

We summarize the key aspects in data preparation:

- **Open Dataset:** The filtered MovieLens 20m dataset contains 137055 users, 1231 movies and 2606513 movie ratings.

- **Target Users:** In total, we select 20 distinct target users from the Movielens dataset. This simulates a real-world scenario where 20 customers are in the store and accessing the Internet via Wi-Fi. In reality, a target user t's number of rated movies $|I_t|$ (with ground truth ratings) may or may not be near the number of recommended movies $|\hat{I}_t|$ (which is set to 50 in the experiments). To conduct experiments on different users, we randomly select 10 targets (T1 to T10) from the set of users whose $|I_t|$ is within the range from 50 to 60. We also randomly select 10 targets (T11-T20), whose $|I_t|$ is in a wider range from 20 to 100. Table 2 summarizes $|I_t|$ of each target user[9].

- **Ground Truth Ratings:** Each target user's ground truth ratings are all ratings he/she assigns in MovieLens. We round decimal ratings to integers based on IEEE 754 standard for arithmetic operations.

- **Movie Titles of Recommendations:** We sort movie titles in the filtered MovieLens by the number of ratings in descending order. For each target user, we randomly select 50 movies from the top-500 as the recommended movie set whose ratings will be predicted by Netflix. Table 2 summarizes $|\hat{I}_t|$ of each target user[9].

- **Predicted Ratings by Netflix:** Netflix displays the predicted rating via a feature shown under the movie title as "Our best guess for ... is ...". We record such ratings for all movies in the recommended movie set. We round decimal ratings to integers based on IEEE 754 standard for arithmetic operations.

5.2 Metrics

In the experiments, Rec2PI infers the target user's preference profile about movies. To evaluate the effectiveness, we need to first obtain the user's inferred behavior (inferred ratings) based on the inferred profile and then compare it to the original behavior (ground truth ratings). For a number of repeated runs (10 in the experiments), the average distance between inferred ratings and ground truth ratings,

[8]http://grouplens.org/datasets/movielens/20m/

[9]For some target users, $|I_t|$ or $|\hat{I}_t|$ differs from the range or value setting because Netflix is constantly updating their streaming movies and some pre-selected movies have become inaccessible when we collect the ratings (Dec. 2015).

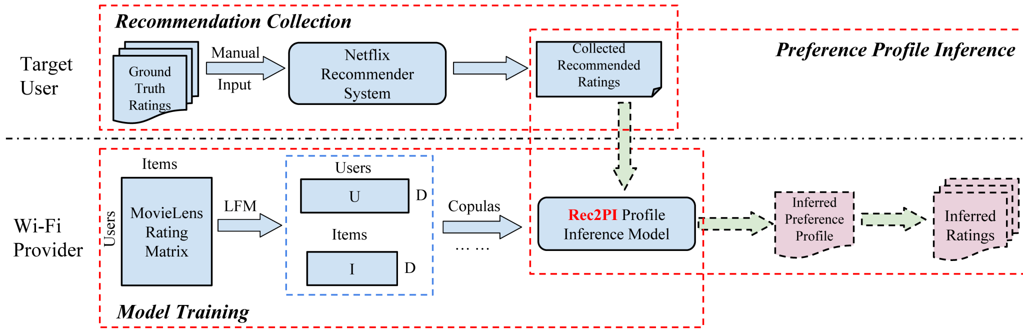


Figure 3: Experimental setup for the evaluation of Rec2PI (Refer to Section 5.1 for details).

Table 2: Statistics of Target Users

TID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
$\lvert I_t \rvert$	55	54	57	49	56	51	44	53	46	46	22	93	22	60	23	21	76	26	73	21
$\lvert \hat{I}_t \rvert$	49	47	49	49	49	47	48	47	39	41	50	50	50	50	50	50	50	50	50	50

from retailers' viewpoint, should not only be small (accuracy), but also remain stable (stability). Therefore, we use the following two metrics:

1. Root mean squared error (RMSE) to measure the distance in each run. RMSE is defined as:

$$\text{RMSE} = \sqrt{\left(\sum_{n=1}^{N}(r_n - r'_n)^2\right)/N}, \tag{15}$$

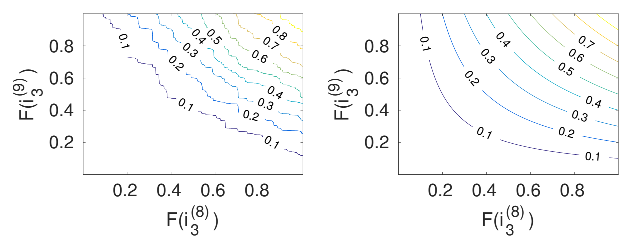
   where N is the number of ground truth ratings, r_n is the ground truth rating and r'_n is the inferred rating. Denote $\text{MEAN}_{\text{RMSE}}$ as the average RMSE value of all repeated runs on each target user.

2. Standard deviation of RMSE values (SD_{RMSE}) to measure the stability.

5.3 Algorithm Settings and Baselines

The types of copulas that are used to fit the data should be specified for the proposed algorithms. For CPM, we specify Gaussian copula in Line 8 of Algorithm 2. For VPM, we specify Clayton copula for modeling the dependence structures for pair copulas constructed by item factors (Line 7 of Algorithm 2). To demonstrate how we specify these copula types, we show the empirical copula result for a certain pair of item factors with a specific rating value. In Figure 7(a), we plot the empirical copula contour of the pair of 8-th and 9-th item factors for 3-valued ratings. There are no well-known copula families that well fit the empirical copula contour. The closest fit that we can find so far is Clayton copula. Therefore, we approximate the empirical copula with Clayton copula. Figure 7(b) depicts the fitted Clayton copula contour. In Subsection 5.4, we show that with the best approximation we have found, we can obtain performance results better than that of existing methods with regard to stability. If in the future, a new parametric copula type can be found to fit the empirical copula better, we expect even better performance results.

To illustrate the effectiveness of the proposed algorithms, we compare them to three baseline methods that estimate a target user t's profile Λ_t in an intuitive way:



(a) Empirical contour (b) Fitted contour

Figure 7: Empirical and fitted Clayton copula contour comparison of the pair of 8-th and 9-th item factors for rating $x = 3$.

1. **Most Similar User (MSU):** We first factorize $\mathbf{R}$ to get $\Lambda_u, \forall u \in U$ and $\Gamma_i, \forall i \in I$. For each $u \in U$, we construct a vector $\mathbf{v}_u$, containing ground truth and predicted (if u has not rated i) ratings for $i \in \hat{I}_t$. Then we find a user $u^* = \underset{u \in U}{\text{argmin}} \lVert \hat{\mathcal{R}}_t - \mathbf{v}_u \rVert$. Set $\Lambda_t = \Lambda_{u^*}$.

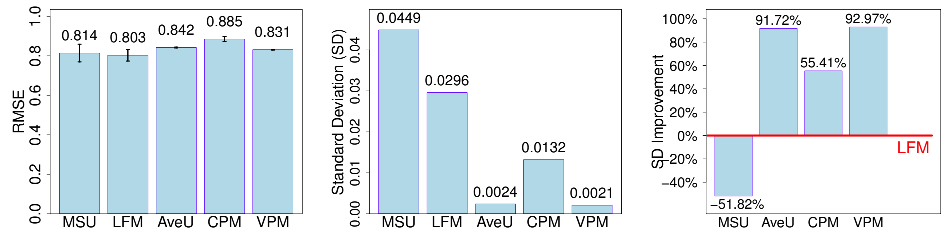
2. **Average User Profile (AveU):** We first factorize $\mathbf{R}$ to get $\Lambda_u, \forall u \in U$ and $\Gamma_i, \forall i \in I$. For each element $t^{(d)} \in \Lambda_t$ (for $1 \le d \le D$), set $t^{(d)} = \frac{1}{|U|} \sum_{u \in U} u^{(d)}$, for $1 \le d \le D$.

3. **LFM:** Treat the recommended ratings as t's ground truth ratings. Add ratings in $\hat{\mathcal{R}}_t$ to $\mathbf{R}$. Then factorize the rating matrix $\mathbf{R} \cap \hat{\mathcal{R}}_t$ to get Λ_t.

The dimension of user/item factors is set as $D = 10$. We use the LFM implementation provided by the authors of [6].

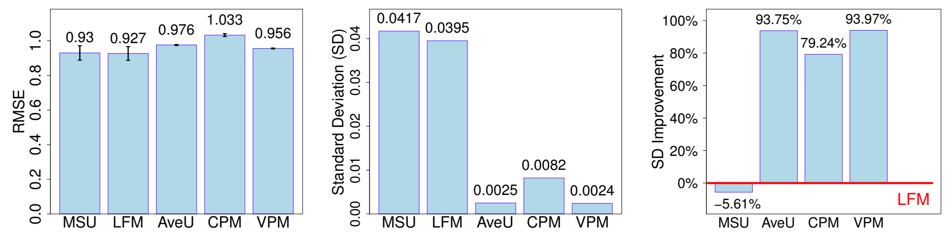
5.4 Performance Comparison

For each target user, we run all algorithms 10 times. We then take the average values of the two metrics, $\text{MEAN}_{\text{RMSE}}$ and SD_{RMSE}, of each group of target users and of all 20 target users, as specified in Subsection 5.1. The results are shown in Figures 4, 5, and 6. Each figure consists of three subfigures, including (a) the average of $\text{MEAN}_{\text{RMSE}}$ over the users, (b) the average of SD_{RMSE} over the users and (c) the SD improvement of each algorithm against LFM, which



(a) Ave($\mathrm{MEAN}_{\mathrm{RMSE}}$) (b) Ave($\mathrm{SD}_{\mathrm{RMSE}}$) (c) SD IMP over **LFM**

Figure 4: Average metrics and SD improvements against LFM of users from T1 to T10.



(a) Ave($\mathrm{MEAN}_{\mathrm{RMSE}}$) (b) Ave($\mathrm{SD}_{\mathrm{RMSE}}$) (c) SD IMP over **LFM**

Figure 5: Average metrics and SD improvements against LFM of users from T11 to T20.

has the lowest average $\mathrm{MEAN}_{\mathrm{RMSE}}$ value. Figures 4 and 5 depict the results of the two target user groups. From the figures, we can see that the average RMSE values do not indicate significant difference for the 5 algorithms, with LFM only slightly better than others. The average SD values and the SD improvements, however, clearly show the advantages of our proposed methods, especially VPM. While CPM and VPM maintain close performance to LFM in accuracy (similar average RMSE values), they achieve much better stability (less variance), with average SD improvements of all target users at 69.05% and 93.54%, respectively, as shown in Figure 6. This implies that VPM is more robust and can generate much stabler profiles. Although AveU is also stable (because it takes the average of all users' factors), it is less accurate than VPM.

In addition, the superiority of our proposed vine-based modelling of dependence structure among item factors is verified due to the fact that VPM improves the performance over CPM (which applies independence assumption for item factors) with regard to both accuracy and stability. The promising performance implies that the inferred target users' ratings by our proposed methods, especially VPM, are consistently close to their ground truth ratings in different runs. This characteristic has an important practical meaning because the user preference inferred by Rec2PI is not only accurate, but also stable. Retailers thus can take advantage of it to maintain reliable and valuable customer management.

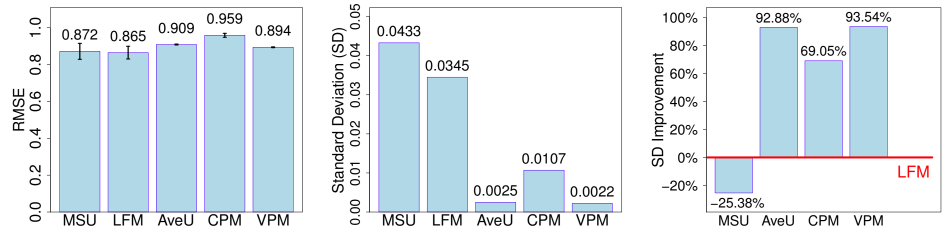
6. RELATED WORK

Rec2PI solves a user profile inference problem arising in Wi-Fi data mining. Wi-Fi providers of brick-and-mortar stores can take advantage of Rec2PI to infer customers' preference based on Wi-Fi collected information so as to obtain a better understanding of in-store customers. Data mining on Wi-Fi collected information has not been fully investigated. Most relevant research works about mining the interaction between users and items focus on online recommender systems (RS) in a variety of domains, such as Ecommerce ([13]), online social networks ([4]) and Internet streaming media ([9]). As Rec2PI does not assume access to a target user's ground truth behavior, it is fundamentally different from these existing works on RS. Rec2PI is also related to transfer learning. Pan et al. [19] categorized and reviewed the progress on transfer learning for classification, regression, and clustering problems. Pan and Yang [20] proposed to model both the numerical ratings and binary auxiliary preference in a principled way, and therefore to alleviate data sparsity (cold start) for collaborative filtering domains expressed in numerical ratings. Wongchokprasitti et al. [21] transferred user models built by one system to another to address the cold start problem. Indeed, transfer learning is part of Rec2PI: Rec2PI learns the pattern of user behavior from open datasets and applies it to the novel recommendation-based user profiling (the middle step in the red dashed box in Figure 1). However, Rec2PI goes beyond transfer learning because inferring a target user's profile from recommended content is different from the inference from his/her ground truth behavior and requires novel and dedicated approaches. To the best of our knowledge, the research problem posed in Rec2PI for Wi-Fi data mining is completely new.

The core of Rec2PI involves a copula-based probabilistic model, which is vital to profile inference. As a powerful statistical tool, copulas (including vines) are quite mature and have been broadly used in the domain of financial analysis for multivariate dependence modelling [1, 8]. The properties of copula theory are described in detail in [17]. With copula theory, we can capture the dependence structure between user factors and item factors, as well as those among item factors.

7. CONCLUSIONS

In this paper, we initiated the study of a novel value-added service, Rec2PI, to Wi-Fi data mining. The goal is to infer



(a) Ave($\text{MEAN}_{\text{RMSE}}$) (b) Ave($\text{SD}_{\text{RMSE}}$) (c) SD IMP over **LFM**

Figure 6: Average metrics and SD improvements against LFM of all target users.

a user's preference profile given a set of recommendations from third-party RS, whose algorithms and the used dataset in the recommendation are unknown.

We formulated the inference task as a marginal maximum a posteriori probability (MAP) estimation problem. We provided a formal definition of Rec2PI in the context of rating-based item recommendation, including the modelling of user/item profile, user behavior and the interaction between users and items. We proposed a novel approach incorporating copulas into the modelling procedure to capture the dependence structure between any user feature and a set of item features. Vines for multivariate copulas capture the dependence structures among item features. We learned the inference model on an open dataset and evaluated the performance of Rec2PI using recommendations generated by a real-world RS. Evaluation results showed that our proposed algorithms, especially VPM, are accurate and stable.

We believe that we can further improve on our results with better parametric copulas. Finding more accurate copula models is a promising future work direction.

8. ACKNOWLEDGMENTS

This research was partially supported by the Natural Sciences and Engineering Research Council of Canada under grant No.283327-2012, No.261409-2013 and No.311999-2011.

9. REFERENCES

[1] K. Aas, C. Czado, A. Frigessi, and H. Bakken. Pair-copula constructions of multiple dependence. *Insurance: Mathematics and economics*, 44(2):182–198, 2009.

[2] K. Azam and M. K. Pitt. *Bayesian inference for a semi-parametric copula-based Markov chain.* University of Warwick. Dept of Economics, 2014.

[3] E. C. Brechmann and U. Schepsmeier. Modeling dependence with c-and d-vine copulas: The r-package cdvine. *J. of Statistical Software*, 52(3):1–27, 2013.

[4] G. Carullo, A. Castiglione, A. De Santis, and F. Palmieri. A triadic closure and homophily-based recommendation system for online social networks. *J. of World Wide Web*, 18(6):1579–1601, 2015.

[5] N. N. Chan, W. Gaaloul, and S. Tata. A web service recommender system using vector space model and latent semantic indexing. In *AINA 2011*, pages 602–609.

[6] C. Chen, L. Zheng, A. Thomo, K. Wu, and V. Srinivasan. Comparing the staples in latent factor models for recommender systems. In *ACM SAC*, pages 91–96, 2014.

[7] P. Embrechts, F. Lindskog, and A. McNeil. Modelling dependence with copulas. *Rapport technique, Département de mathématiques, Institut Fédéral de Technologie de Zurich, Zurich*, 2001.

[8] P. Embrechts, F. Lindskog, and A. McNeil. Modelling dependence with copulas and applications to risk management. *Handbook of heavy tailed distributions in finance*, 8(1):329–384, 2003.

[9] C. A. Gomez-Uribe and N. Hunt. The netflix recommender system: Algorithms, business value, and innovation. *ACM Trans. Manage. Inf. Syst.*, 6(4):13:1–13:19, Dec. 2015.

[10] H. Joe. Families of m-variate distributions with given margins and m (m-1)/2 bivariate dependence parameters. *Lecture Notes-Monograph Series*, pages 120–141, 1996.

[11] Y. Koren and R. Bell. Advances in collaborative filtering. In *Recommender systems handbook*, pages 145–186. Springer, 2011.

[12] K. Lakiotaki, N. F. Matsatsinis, and A. Tsoukias. Multicriteria user modeling in recommender systems. *IEEE Intelligent Systems*, 26(2):64–76, Mar. 2011.

[13] G. Linden, B. Smith, and J. York. Amazon. com recommendations: Item-to-item collaborative filtering. *Internet Computing, IEEE*, 7(1):76–80, 2003.

[14] H. Ma, H. Yang, M. R. Lyu, and I. King. Sorec: Social recommendation using probabilistic matrix factorization. In *CIKM 2008*, pages 931–940.

[15] S. E. Middleton, N. R. Shadbolt, and D. C. De Roure. Ontological user profiling in recommender systems. *ACM Trans. on Info. Sys.*, 22(1):54–88, 2004.

[16] A. Mnih and R. R. Salakhutdinov. Probabilistic matrix factorization. In *NIPS 20*, pages 1257–1264. Curran Associates, Inc., 2008.

[17] R. B. Nelsen. *An introduction to copulas*, volume 139. Springer Science & Business Media, 2013.

[18] R. Nelson. *An Introduction to Copulas.* Springer, 2006.

[19] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Trans. on Knowl. and Data Eng.*, 22(10):1345–1359, Oct. 2010.

[20] W. Pan and Q. Yang. Transfer learning in heterogeneous collaborative filtering domains. *Artificial intelligence*, 197:39–55, 2013.

[21] C. Wongchokprasitti, J. Peltonen, T. Ruotsalo, P. Bandyopadhyay, G. Jacucci, and P. Brusilovsky. User model in a box: Cross-system user model transfer for resolving cold start problems. In *International Conference on User Modeling, Adaptation, and Personalization*, pages 289–301. Springer, 2015.