Proceedings of the Ninth International AAAI Conference on Web and Social Media

Leveraging Product Adopter Information from Online Reviews for Product Recommendation

Jinpeng Wang[1], Wayne Xin Zhao[2], Yulan He[3], and Xiaoming Li[1]
[1]School of Electronic Engineering and Computer Science, Peking University, China
[2]School of Information, Renmin University of China
[3]School of Engineering & Applied Science, Aston University, UK
{JooPoo, lxm}@pku.edu.cn, batmanfly@gmail.com, y.he@cantab.net

Abstract

The availability of the sheer volume of online product reviews makes it possible to derive implicit demographic information of product adopters from review documents. This paper proposes a novel approach to the extraction of *product adopter mentions* from online reviews. The extracted product adopters are then categorise into a number of different demographic user groups. The aggregated demographic information of many product adopters can be used to characterise both products and users, which can be incorporated into a recommendation method using weighted regularised matrix factorisation. Our experimental results on over 15 million reviews crawled from JINGDONG, the largest B2C e-commerce website in China, show the feasibility and effectiveness of our proposed framework for product recommendation.

Introduction

With the increasing popularity of online e-commerce services, more and more people buy products online. As such, a large volume of online reviews have been constantly generated by users. Since review data contain rich information about users' feedback and opinions towards products they purchased, mining online reviews has attracted much interest (Hu and Liu 2004) which could be subsequently used for product sales prediction (Liu et al. 2007). Nevertheless, we argue that online reviews sometimes also contain implicit user demographic information which could be leveraged for product recommendation. Consider the following review sentence about "`iPhone 6`":

"`I bought my son an iPhone 6, and he was very happy with the new phone.`"

Although we don't know the age or sex of the author who wrote this review, we can infer from the above sentence that the actual adopter of "`iPhone 6`" is a young male. Here, we call the phrase "`my son`" as the *product adopter mention*. In this example, product purchaser and adopter are not the same person. However, the explicit mention of *product adopter* reveals the demographic information of the actual user of the product. If we can gather more demographic information of users who adopted a specific product, then we can easily get a picture of its target audience and can subsequently make a better product recommendation. For example, if another user expressed a purchase intent, "`I want to buy a smart phone for my son. Any suggestions?`" In such a case, "`iPhone 6`" might be a good candidate for recommendation.

Traditionally, marketing research often relies on surveys or questionnaires to collect customers' demographic profiles or other user-related information in order to perform market segmentation or identify target audience (Cohen 2005). Such an approach is very time-consuming and also suffers from the sampling bias problem. On the contrary, online reviews are readily available and make it possible to automatically derive users' demographic information. In this paper, we aim to extract *product adopter mentions* from online reviews, categorise product adopters into a number of different user groups, aggregate demographic information of many adopters to form product demographics, which can be used for future product recommendation. To the best of our knowledge, the idea of mining product adopter information from reviews and use it for product recommendation has never been explored in a large dataset before.

To leverage product adopter information for recommendation, we are facing two major challenges. First, review data are very noisy and often contain slang, misspellings and emoticons (Pang and Lee 2008), which make it difficult to extract product adopter mentions reliably at a large scale. Second, with the extracted product adopters, it is not straightforward to model such information and incorporate it into traditional product recommendation methods. To address the first challenge, we develop an unsupervised bootstrapping method to automatically derive patterns to extract product adopter mentions from a large review dataset constructed based on the largest B2C e-commerce website in China, JINGDONG[1]. Using the automatically derived product adopter extraction patterns, we have found that more than 10% of the reviews contain at least one adopter mention. The adopter mentions are grouped into six categories using the idea of demographic segmentation in marketing research. Product demographic is then represented by a dis-

[1]http://www.jd.com/

tribution over six such categories, which is called *product adopter distribution*. Similarly, each user is represented by its purchase preference patterns (whom she bought a product for) over the same six categories, called *user preference distribution*. We have found that: 1) A significant number of products and users have their respective product adopter distributions peaked on one or two adopter categories; 2) Users in different adopter categories have quite distinct product preference patterns. We then propose a recommendation method based on weighted regularised matrix factorisation by incorporating the derived distributions charactering products and users, which has effectively addressed the second challenge. Our experimental results show that the proposed method outperforms a number of competitive baselines consistently.

Related Work

Product recommendation plays an important role to improve the sales of e-commerce companies. Early work on product recommendation typically uses collaborative filtering (CF) to make recommendations based on matching users with similar "tastes" or interests which can be revealed through users' past purchase behaviors or rating patterns (Sarwar et al. 2001; Adomavicius and Tuzhilin 2005; Linden, Smith, and York 2003). Recommender systems based on CF suffers from data sparseness of the involved user-item matrix that two random users might not have rated any item in common and thus it is difficult to compute the similarity between them. In recent years, there have been increasing interests in exploring social networks for product recommendation in order to alleviate the data sparseness problem since users who are connected are likely to have similar preferences. One type of social relations is trust relation which is established when users explicitly state how much they consider other users trustworthy, for example, on the Epinion[2] product review site. Methods have been proposed to propagate trust values in trust networks and incorporate the trust information in CF-based recommendation framework (Massa and Avesani 2007; Jamali and Ester 2009; Tang et al. 2012). Ma et al. (2011) proposed a factor analysis approach based on probabilistic matrix factorization by incorporating social context and social tags. Symeonidis et al. (2011) argued that users can form implicit social rating networks through co-commenting on the same products or similarly co-rating products. Users can also build explicit social network by adding each other as friends. They proposed a method which combines multiple similarity matrices derived from explicit or implicit social networks for product recommendation and rating prediction.

We do not intend to explore social network information for product recommendation. Instead, we aim to develop a recommender system based on inferred user demographic information from online reviews. Demographic-based recommendation approaches typically utilize the demographic information obtained directly from user websites (Pazzani 1999; Giering 2008) or questionnaires (Lekakos and Giaglis 2007; Qiu and Benbasat 2010) for rating prediction or product recommendation. Seroussi et al. (2011) proposed to extract topics from user-generated text using the Latent Dirichlet Allocation (LDA) model, termed as text-based user attributes, in addition to demographic attributes explicitly supplied by users. Both types of attributes were then integrated into a matrix factorization model for rating prediction. One problem here is that the latent topics extracted by LDA do not necessarily correspond to user-specific attributes. Korfiatisa and Poulos (2013) proposed to build a demographic recommender system by extracting service quality indicators (star ratings) and consumer types from hotel reviews. In particular, they defined different demographic groups by consumer types through unsupervised clustering of online reviews based on the assumption that different types of travelers assess each quality indicator differently and are thus not necessarily depending on their cultural or age background. Since there might be millions of reviews generated each day, clustering reviews in order to derive users' demographic groups can be extremely computationally expensive.

With the rapid growth of online e-commerce services, online review mining has become a hot research topic (Pang and Lee 2008). In particular, it has been shown that online review are useful to improve the results of product ranking or recommendation. Liu et al. (2007) proposed to use a sentiment model to predict sales performance; while in (McGlohon, Glance, and Reiter 2010), composite rating scores were derived from aggregated reviews collected from multiple websites using different statistic- and heuristic-based methods and were subsequently used to rank products and merchants. Ganu et al. (2009; 2013) derived text-based ratings of item aspects from review text and then grouped similar users together using soft clustering techniques based on the topics and sentiments that appear in the reviews. In short, the core idea of these studies is to transform opinionated text into sentiment scores, which can be used to rank products.

Instead of relying on sentiment mined from online reviews for product ranking or recommendation, we propose to infer users' demographic information from product adopter mentions in review text, which can be subsequently incorporated into standard recommendation algorithms such as matrix factorization to achieve a better recommendation results. As opposed to existing demographic-based recommender systems, we do not require the demographic profile to be explicitly supplied by users. In fact, in many e-commerce websites such as JINGDONG, users are not required to fill in their demographic attributes which makes it impossible to obtain such information directly. Furthermore, we focus on the situation where a product buyer is different from a product adopter, i.e., a user bought a product for others. In this case, even with the availability of product buyers' registered demographic information, it is no longer useful for the recommendation task. Thus, we need to infer the demographic information of the actual product adopters for efficient product recommendation. To the best of our knowledge, dealing with the mismatch between product buyers and adopters for automatic inference of demographic information from text and subsequently incorporate it for product recommendation have never been studied before.

[2]http://www.epinions.com

Data Collection

We construct a large review dataset from the largest B2C e-commerce website (Amazon-like) in China, JINGDONG, by crawling 139 million reviews of 0.2 million products from 12 million users. The data span one year from January 1, 2013 through December 31, 2013. We use the toolkit Jieba[3] to segment Chinese character streams into words.

JINGDONG has a policy that only users who have bought a product are allowed to write reviews about that product. In another words, a review document would correspond to a unique transaction record. Also, each review has a user ID. This allows us to build a test set to evaluate our product recommendation method which will become clear in the Experiments section. Although our research was conducted on the dataset constructed from JINGDONG, the methods proposed here are equally applicable to reviews collected from other websites.

Extracting Demographic Information from Reviews

In this section, we first present a novel bootstrapping approach for the extraction of product adopter mentions from review documents, and then discuss how to categorise the extracted product adopters into six user categories based on the idea of demographic segmentation in marketing research. We subsequently use the product adopter categories to characterise both products and users in the form of *product adopter distribution* or *user preference distribution*. We perform an initial data analysis to find out whether we observe a significant number of products or users having their respective *product adopter distribution* or *user preference distribution* peaked on a small number of adopter categories; and whether users tend to buy products which share similar demographic features.

Extraction of Product Adopter Mentions

A bootstrapping-based extraction method The sheer volume of review data and the informal writing styles observed in many reviews make it infeasible to develop supervised methods relying on annotated data to extract product adopter mentions at a large scale. As such, we resort to unsupervised methods. We notice that some product adopter mentions could be described by the same linguistic pattern. For example, in a sentence "`I bought my son this phone`", the phrase "`my son`" is the adopter of the phone. If we can learn the pattern "`buy somebody something`", then we can extract the corresponding product adopter mention. We propose a bootstrapping approach in Algorithm 1 to iteratively learn the patterns and extract adopter mentions. The approach starts with some seed patterns such as "`buy somebody something`" and "`a gift to somebody`". In each iteration, we apply existing patterns to extract new adopter mention phrases with the function $\mathsf{ExtractAdopterMentionPhrases}(\cdot,\cdot)$, and then learn new patterns with the extracted phrases with the functions $\mathsf{GeneratePatterns}(\cdot,\cdot)$ and $\mathsf{ExtractTopFrequentPatterns}(\cdot)$. To generate patterns with $\mathsf{GeneratePatterns}(\cdot,\cdot)$, for each adopter mention phrase, we first extract the preceeding n_1 tokens and the following n_2 tokens, and then combine the $(n_1 + n_2)$ tokens as the candidate patterns. We have found that many single-token patterns are noises, and patterns with more than two tokens yield little improvement when applied on short review text in our experiments. As such, we only consider two-token patterns and require $(n_1 + n_2)$ to be 2, where $0 \le n_1 \le 2$ and $0 \le n_2 \le 2$.

We also propose a pattern filtering step (Lines $14-17$) to reduce spurious patterns that may lead to a large number of false positives, i.e., incorrect product adopter mentions. The main idea is that the adopter mention phrases identified by a good pattern should not deviate from the previously identified phrases too much. Here we use the Jaccard coefficient to measure the similarity among the extracted phrases and empirically set the threshold δ to 0.3.

Extraction results At the end of the iterative bootstrapping process, we have derived 45 patterns. Some learned patterns are listed in Table 1. We invited two human judges to conduct manual checking. As it is infeasible to check all the extracted mentions, we only consider those with occurrence frequency larger than 30 and are left with 410 mentions. A total of 363 mentions are judged to be correct, which gives a precision of 88.5% . We have found that without the pattern filtering step (Lines $14-17$ in Algorithm 1), the extraction accuracy dropped to 63.5%. This shows the importance of reducing false positives using our proposed filtering method. We also notice that some extracted mentions do not carry clear demographic information (e.g., "`someone`" or "`my family`"). As such, we further remove 104 ambiguous mentions and our final list contains 259 adopter mentions.

Statistics analysis Using our proposed bootstrapping approach for the extraction of product adapter mentions, we found that 10.8% out of a total of 139 million reviews contain at least one adopter mention. This is equivalent to 15 million reviews. Since it is impossible to enumerate all the possible linguistic patterns describing adopter mentions, this value can be considered as a lower bound of the proportion of review documents containing product adopter mentions. We further plot in Figure 1 the distribution of adopter mentions versus products and users respectively. It can be observed that more than 48.5% products (receiving the reviews) and 25.4% users (posting the reviews) have at least 10 reviews containing adopter mentions. The above results show that the product adopter information indeed prevails in review documents.

Grouping Adopter Mentions into Categories

In online reviews, the same entity can be referred to in many different ways. For example, "`mum, mom, mother`" all refer to the same entity "`mother`". In addition, some different adopter mentions may share similar demographic features. For example, "`grandpa, father-in-law`" are all males and possibly over age 55. As such, it makes sense to group product adopter mentions into different categories

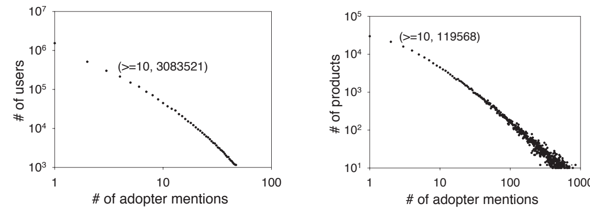
[3] https://github.com/fxsjy/jieba

Algorithm 1: Bootstrapping algorithm for extracting product adopter mentions from online reviews.

Input: review sentence corpus $\mathcal{R}$, seed extraction patterns $\mathcal{P}^{(seed)}$
Output: an set of learned extraction patterns $\mathcal{P}$ and a set of extracted adopter mentions $\mathcal{R}$;
$\mathcal{P} \leftarrow \mathcal{P}^{(seed)}, \mathcal{P}' \leftarrow \mathcal{P}^{(seed)}$;
$\mathcal{R}' \leftarrow \emptyset, \mathcal{R} \leftarrow \emptyset$;
repeat
  $\mathcal{R}' \leftarrow \emptyset$;
  for $each\ pattern\ p \in \mathcal{P}'$ **do**
    $\mathcal{R}_p \leftarrow \emptyset$;
    for $each\ sentence\ s \in \mathcal{R}$ **do**
      if *p exists in s* **then**
        $\mathcal{R}_p \leftarrow \mathcal{R}_p \cup$ ExtractAdopterMentionPhrases (p,s);
      end
    end
    if *Jaccard*$(\mathcal{R}_p, \mathcal{R}) \leq \delta$ *and* $p \notin \mathcal{P}^{(seed)}$ **then**
      $\mathcal{R}_p \leftarrow \emptyset$;
      Remove p from $\mathcal{P}'$;
    end
    $\mathcal{R}' \leftarrow \mathcal{R}_p \cup \mathcal{R}'$;
  end
  $\mathcal{P} \leftarrow \mathcal{P} \cup \mathcal{P}', \mathcal{R} \leftarrow \mathcal{R} \cup \mathcal{R}'$;
  $\mathcal{R}' \leftarrow \emptyset, \mathcal{P}' \leftarrow \emptyset$;
  for $each\ sentence\ s \in \mathcal{R}$ **do**
    for $each\ demographic\ phrase\ m \in \mathcal{R}'$ **do**
      $\mathcal{P}' \leftarrow \mathcal{P}' \cup$ GeneratePatterns(s, m);
    end
  end
  $\mathcal{P}' \leftarrow$ ExtractTopFrequentPatterns$(\mathcal{P}')$;
until *No new pattern is identified*;
return *An set of learned extraction patterns* $\mathcal{P}$ *and a set of extracted adopter mentions* $\mathcal{R}$*;*

Example extraction patterns for product adopter mentions
Seed: 给...买(buy something for ...)
Extracted patterns: 送...用(a gift for ... to use), 帮...代购(order something for ...), 送给...了(a gift for ...), 帮...买(buy something for ...), 帮...代买(buy something for ...), 帮...订(order something for ...), 帮...购买(purchase something for ...), 希望...喜欢(hope ... like it), 给...吃(a gift for ... to eat), 买给...玩(a gift for ... to play)

Table 1: Some learned extraction patterns for product adopter mentions.



(a) #mentions v.s. #users (b) #mentions v.s. #products

Figure 1: The distribution of adopter mentions for users and products.

where each category shares similar demographic information. To do this, we first remove ambiguous mentions such as "others" and "people" and only keep the mentions with a relatively clear demographic profile. Following the idea in *market segmentation*[4], we mainly consider two types of demographic characteristics, age and sex. We have invited two senior e-commerce officers in charge of market promotion from JINGDONG to give us advices on categorising product adopter mentions. At the end, we have identified five major categories relating to the mention of relatives based on age and sex: *Children*, *Young female*, *Old female*, *Young male* and *Old male*. Apart from these five categories relating to relatives, we have further identified a category relating to *Colleagues*. Although it is difficult to identify clear demographic features from this category, it is observed that some office products such as printers and fax machines are more closely related to *Colleagues* than other categories. As such, the *Colleagues* category is also taken into account and we have a total of six user categories. Table 2 shows some example adopter mentions and their corresponding categories. The full categorization information can be found in http://goo.gl/avne1y.

It is worth noting that for some adopter mentions, it is difficult to accurately classify them into demographic attribute categories. For example, for the product adopter mention "wife", it is hard to estimate the actual age range and we vaguely classify it into the *Young female* category.As will be shown later, grouping adopter mentions into demographic categories and subsequently estimating product preference probabilities of each category has the smoothing effect. The product preference probabilities are averaged over many "pseudo-labeled" product adopter mentions which makes the estimated probabilities more tolerant to the demographic mapping errors.

Product and User Characteristics Learning

With the aforementioned six categories of product adopters and let $\mathcal{C}$ be the set of categories of adopter mentions, we can characterise products and users in the following ways:
Product adopter distribution: Given a product e, the *product adopter distribution* is a vector with six elements where each element is a proportion of the mentions in category $c \in \mathcal{C}$ in the reviews about product e. This distribution es-

[4]http://en.wikipedia.org/wiki/Market_segmentation#Demographic_Segmentation

Category	Example product adopter mentions
Children	小孩(kids) 她小孩(her kid) 新生儿(new-horn baby) 婴儿(baby) 小侄女(little niece) 小外甥(little nephew) 儿童(children) 孙子(grandson)
Young female	妹妹(younger sister) 姐姐(elder sister) 表妹(cousin) 女朋友(girlfriend) 媳妇(daughter-in-law) 老婆(wife) 弟媳(younger sister-in-law) 孕妇(pregnant woman)
Old female	妈妈(mother) 老妈(mother) 阿姨(aunt) 姑姑(aunt) 岳母(mother-in-law) 丈母娘(mother-in-law) 奶奶(grandmother)
Young male	弟弟(younger brother) 哥哥(elder brother) 表弟(cousin) 哥们(buddy) 兄弟(brothers) 丈夫(husband) 男朋友(boyfriend)
Old male	爸爸(father) 老爸(father) 叔叔(uncle) 舅舅(uncle) 岳父(father-in-law) 公公(father-in-law) 爷爷(grandfather) 姥爷(grandpa)
Colleagues	公司(company) 办公室(office) 员工(employee) 分公司(branch) 厂里(factory) 工人(worker) 部门(department)

Table 2: Example product adopter mentions and their corresponding categories.

sentially characterises the demographics of product e by the users who have actually used the product. We estimate the conditional probability that the adopter of product e belongs to category c, $P(c|e)$, using maximum likelihood

$$P(c|e) = \frac{\#N(e,c) + \gamma}{\#N(e,\cdot) + |\mathcal{C}|\gamma}, \tag{1}$$

where $\#N(e,c)$ is the number of reviews about product e which contain adopter mentions in category c. We use Laplace smoothing to avoid the zero probability which might be caused by data sparsity.

User preference distribution: Similar to the definition of product adopter distribution, *user preference distribution* characterises a user's purchase preference pattern (whom she bought a product for) over the six adopter categories. Thus, the *user preference distribution* is also a six-element vector with each of its element $P(c|u)$, the probability of user u who tends to buy products for people in category c, defined below

$$P(c|u) = \frac{\#N(u,c) + \gamma}{\#N(u,\cdot) + |\mathcal{C}|\gamma}, \tag{2}$$

where $\#N(u,c)$ is the number of reviews written by u which contain adopter mentions in category c.

It can be understood that *product adopter distribution* and *user preference distribution* characterize the preference over the six adopter categories from the perspective of products and users respectively. *Product adopter distribution* captures product demographics which are represented by a collection of the characteristics of the people who have actually adopted that product; while *user preference distribution* measures the preference level of a user over people in different adopter categories, that is, whom the user is more likely to buy products for.

Category-specific product distribution: We would expect that users in the same category prefer similar products and therefore introduce the *category-specific product distribution* to characterize the categorical preference over products. Let $\mathcal{E}$ be the set of all the candidate products in our dataset. Given a category c, the *category-specific product distribution* is a vector with $|\mathcal{E}|$ elements where each element is a proportion of the reviews about product e which contains a mention of category $c \in \mathcal{C}$. This distribution essentially characterises the popularity of product e in the adopters of category c. We estimate the conditional probability that a product e given category c, $P(e|c)$, using maximum likelihood

$$P(e|c) = \frac{\#N(e,c) + \mu}{\sum_{e' \in \mathcal{E}} \#N(e',c) + |\mathcal{E}|\mu}. \tag{3}$$

#HP Categories	0	1	2	3
Users	0.94%	86.98%	12.08%	0
Products	8.15%	83.80%	8.05%	0

Table 3: The proportions of users and products with respective to different numbers of highly-preferred categories.

Initial Data Analysis

In the first set of experiment, we would like to find out the shapes of product adopter distributions and user preference distributions. In particular, we want to find out whether we can observe non-uniform patterns in both distributions, i.e. the distribution is peaked at very few adopter categories. For example, if a certain brand of shoes is typically used by the older males population, then it should be recommended to users who want to buy shoes for people in the *Old male* category. Similarly, if someone purchased products for her father most of the time, then it would be helpful to identify such a preference and recommend products accordingly.

We define a *highly-preferred category* if it is assigned with a probability which is at least twice of the uniform probability, i.e. $\frac{2}{|\mathcal{C}|}$. Table 3 presents the proportion of users and products with respective to different numbers of highly-preferred categories. It can be observed that about 90% of the users and products have exactly one or two highly-preferred adopter categories, which indicates most of the products (or users) have rather biased product adopter distribution (or user preference distribution).

In the second set of experiments, we would like to find out whether there is litter overlap among the top ranked products

Categories	*CH*	*YF*	*OF*	*YM*	*OM*	*CO*
CH	-	0	0	0	0	0
YF	0	-	13	7	5	15
OF	0	13	-	2	33	2
YM	0	7	2	-	32	26
OM	0	5	33	32	-	7
CO	0	15	2	26	7	-

Table 4: The number of overlapping products in the top 100 positions of two different categories. *CH*, *YF*, *OF*, *YM*, *OM* and *CO* denote *children*, *young female*, *old female*, *young male*, *old male* and *collegue* respectively.

by category-specific product distributions between two user categories. We cross compare the top 100 products ranked by category-specific product distributions for each category pair and present the results in Table 4, which is essentially a symmetric matrix. Most category pairs have very few products in common. In particular, the *Children* category does not share any common popular products with the other categories. Only 3 category pairs have between 26 and 33 products in common out of the top 100 products, i.e. (*OF*,*OM*), (*YM*, *OM*) and (*YM*, *CO*). This is perhaps not surprising since old people (either male or female) have some product preferences in common (e.g. health products) and men (either young or old) prefers certain products (e.g., electric shaver). Office products which are typically purchased by users in the *Colleagues* category are also preferred by the *Young male* category (e.g., external hard disk or USB stick). For a better understanding of these results, we present the top 10 products ranked by category-specific product distributions for each category in http://goo.gl/avne1y.

In the third set of experiments, we would like to find out how product adopter distributions and user preference distributions change over time. We split our data into T time epochs. Let θ_u and ϕ_e denote the preference distribution of user u and adopter distribution of product e respectively. Furthermore, let $\theta_u^{1,t}$ and $\phi_e^{1,t}$ denote the estimated distribution values using the data from the first to the t-th epoch. We use the Kullback-Leibler divergence[5] $D_{KL}(\theta_u^{1,T}||\theta_u^{1,t})$ and $D_{KL}(\phi_e^{1,T}||\phi_e^{1,t})$ to measure the divergence degree of the estimated distributions on data accumulated up to the t-th time epoch with respective to the last time epoch. Smaller KL divergence values indicate that the two distributions are more similar. In our experiments, we split the data of the entire year of 2013 into six equal time epochs ($T = 6$) with each epoch containing two months' data.

We present the average divergence degree of users and products in Figure 2. It can observed that product adopter distributions have smaller KL divergence values and converge more quickly than user preference distributions. This implies that a product is usually aimed at a specific target audience, as such it has clear demographic characteristics and its product adopter distribution remains relatives stable. While users can choose to buy products for people in different age or sex groups, the user preference distributions therefore appear to be more dynamic.

[5]http://en.wikipedia.org/wiki/Kullback-Leibler_divergence

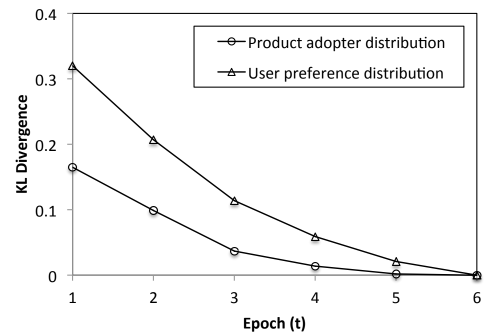


Figure 2: The average divergence degree of product adopter distributions and user preference distributions.

Incorporating the Preference of Adopters into Product Recommendation

Product recommender systems have been shown to boost the sales and increase user satisfaction in e-commerce websites. Typical recommendation techniques include user/item-based collaborative filtering (Sarwar et al. 2001; Adomavicius and Tuzhilin 2005; Linden, Smith, and York 2003) and matrix factorization (Koren, Bell, and Volinsky 2009). Our initial analysis reveals that both users and products have shown some adopter-related patterns: (a) a user indeed has a biased distribution over the six adopter-related categories and (b) an adopter-related category has its own popular products with the category-specific characteristics.

In this section, we propose to incorporate the above intuitions into a traditional recommendation framework to improve the performance of product recommendation. In particular, we choose the commonly used matrix factorization technique as the baseline since it has been shown to be very effective in many recommendation applications.

Estimating the User-Product Conditional Preference Probabilities

The key to our approach is to estimate user-product conditional preference probabilities, which measure the confidence that a user would purchase a product based on the adopter-related information. Let $P(e|u)$ be the conditional preference probability of product e given user u. Here we take a two-step generative approach to learn $P(e|u)$: when a user wants to purchase a product, she will first select an adopter-related category according to her preference distribution; then select a product according to the category-specific product distribution. Formally, we have

$$P(e|u) = \sum_{c \in \mathcal{C}} P(c, e|u) = \sum_{c \in \mathcal{C}} P(e|c)P(c|u), \qquad (4)$$

where $P(e|c)$ and $P(c|u)$ are estimated respectively by Eq. 3 and Eq. 2. Here we assume product e is independent of user u given category c. The derived user-product conditional probabilities directly reflect the preference degree of a user over the candidate products.

Preference Biased Matrix Factorization (PBMF)

Matrix factorization (MF) is one of the most widely used recommendation techniques in product recommendation. Formally, assume that there are N users and M products. A typical latent MF model associates each user u with a user-factor vector $\mathbf{x}_u \in \mathbb{R}^K$, and each product e with an item-factor vector $\mathbf{y}_e \in \mathbb{R}^K$, where K is the number of the latent dimensions. The model aims to fit a known purchase record $r_{u,e}$, which is a binary variable, by a low-dimension approximation $\hat{r}_{u,e} = \mathbf{x}_u^\top \mathbf{y}_e$. This can be solved by minimizing the overall fitting loss

$$\min_{\mathbf{x}_\cdot, \mathbf{y}_\cdot} \Big\{ \sum_{r_{u,e} \text{ is known}} (r_{u,e} - \mathbf{x}_u^\top \mathbf{y}_e)^2 + \lambda(\| \mathbf{x}_u \|^2 + \| \mathbf{y}_e \|^2) \Big\}.$$

The above model assumes an equal confidence level for all the entries. Inspired by (Hu, Koren, and Volinsky 2008), we propose a preference biased matrix factorization (PBMF) approach

$$\min_{\mathbf{x}_\cdot, \mathbf{y}_\cdot} \Big\{ \sum_{r_{u,e} \text{ is known}} c_{u,e}(r_{u,e} - \mathbf{x}_u^\top \mathbf{y}_e)^2 + \lambda(\| \mathbf{x}_u \|^2 + \| \mathbf{y}_e \|^2) \Big\}. \tag{5}$$

where we explicitly incorporate the confidence level of each entry. A direct way is to set $c_{u,e}$ to the user-product conditional probability, i.e. $c_{u,e} = P(e|u)$. However, we notice that there is a significant scale difference between the conditional probabilities and the loss penalty values. As such, we follow the idea in (Hu, Koren, and Volinsky 2008) to set our confidence levels

$$c_{u,e} = \begin{cases} 1 + \alpha \log(1 + P(e|u)), & r_{u,e} = 1, \\ 0, & r_{u,e} = 0. \end{cases} \tag{6}$$

where α is a tuning parameter which controls the weight of the user-product prior probabilities. We perform Laplace smoothing when calculating $P(e|u)$ inside the log function to make sure $c_{u,e}$ is positive.

Parameter Learning In the preference biased matrix factorization model (PBMF), we have two sets of parameters to learn: $\{\mathbf{x}_u\}$ and $\{\mathbf{y}_e\}$. By differentiating Eq. 5 respective to $\mathbf{x}_u$ and $\mathbf{y}_e$, we can obtain the analytic expressions for update

$$\begin{aligned} \mathbf{x}_u &= (\mathbf{Y}^\top \mathbf{D}^u \mathbf{Y} + \lambda \mathbf{I}) \mathbf{Y}^\top \mathbf{D}^u \mathbf{r}_u, \\ \mathbf{y}_e &= (\mathbf{X}^\top \mathbf{D}^e \mathbf{X} + \lambda \mathbf{I}) \mathbf{X}^\top \mathbf{D}^e \bar{\mathbf{r}}_e, \end{aligned}$$

where $\mathbf{D}^u$ and $\mathbf{D}^e$ are two diagonal matrice in which we have $d^u_{ee} = d^e_{uu} = c_{u,e}$, $\mathbf{r}_u$ and $\bar{\mathbf{r}}_e$ are formed by the corresponding row and column vectors of the binary record matrix $\mathbf{R}$ respectively. In each iteration, we alternatively update $\mathbf{X}$ and $\mathbf{Y}$ using the method in (Hu, Koren, and Volinsky 2008), which has the time compexity $\mathcal{O}(K^3N + K^3M + K^2 * L)$ (L is the number of non-zero entries in the purchase record matrix $\mathbf{R}$).

Experiments

In this section, we present the experimental results on product recommendation.

Construction of the Test Collection

Recall that we have collected a total of 130 million reviews from 0.2 million users. We only keep the reviews which contain the explicit mentions of product adopters. Each review comes with an explicit userID which corresponds to a unique transaction record. We first group reviews by userID and obtain a list of the purchased products for each user. We remove the users with fewer than 20 reviews and products with fewer than 50 reviews. After the above filtering step, we build a user-product matrix and set the entry to one if the product is purchased by the corresponding user. We further iteratively remove the users and products with fewer than 10 non-zero entries. Finally, we obtained a test collection of 9,993 users and 19,894 products with a total number of 1,051,594 purchase records. Its sparsity degree is about 99.47%. In order to simulate the real scenario of online product recommendation, we split the data collection into training and test datasets by sorting the purchase records by timestamps. For each user, we take the first $\delta\%$ of her purchase records as the training data, and the remaining $(100 - \delta)\%$ as the test data. To examine the performance with varying amount of training data, we set δ to 50, 66 and 75, which correspond to the $\frac{\#training}{\#test}$ ratios of 1:1, 2:1 and 4:1 respectively.

Methods to Compare

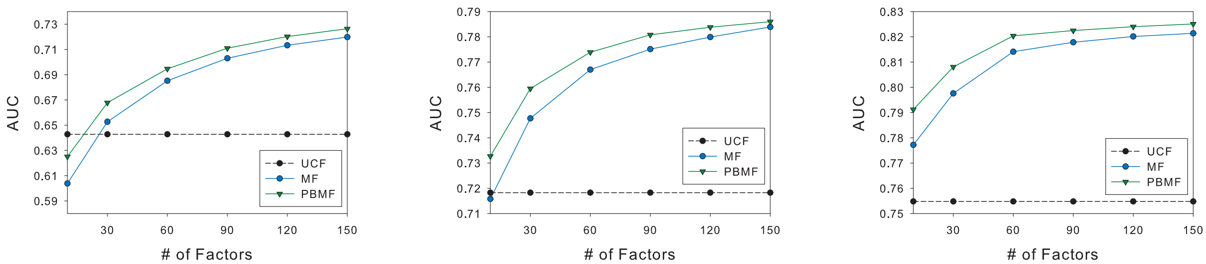
We consider the following methods for performance comparison:

- **Most Popular (MP)**: Items are weighted by how often they have been purchased in the past.
- **User-based Collaborative Filtering (UCF)**: user-based collaborative filtering predicts a test user's interest in a test item based on the rating information from similar users (Adomavicius and Tuzhilin 2005). We also consider incorporating the item-based collaborative filtering (ICF) (Linden, Smith, and York 2003; Sarwar et al. 2001) as a comparison. In our experiments, ICF produced similar results as UCF, therefore we only report the results of UCF. We set the maximum number of neighbors to 50.
- **Matrix Factorization (MF)**: the standard MF method as in (Koren, Bell, and Volinsky 2009).
- **Preference Biased Matrix Factorization (PBMF)**: our proposed method (Eq. 5).

All the methods were implemented using an open source toolkit *MyMediaLite*[6].

For our proposed approach, we have a few parameters to set. The first two parameters are the smoothing parameters used in the estimation of the distributions: γ in Eq. 1 and 2, and μ in Eq. 3. We tuned both parameters on a held-out dataset and found that setting both of them to small values

[6] http://www.mymedialite.net/



(a) $\frac{\#training}{\#test}$=1:1 (b) $\frac{\#training}{\#test}$=2:1 (c) $\frac{\#training}{\#test}$=4:1

Figure 3: Results by varying the number of latent factors.

lead to good and stable performance. We thus set γ to $1e^{-3}$ and μ to $1e^{-4}$. For the regularization coefficient λ in matrix factorization based methods and the tuning parameter α in $c_{u,e}$, we use the default settings in *MyMediaLite*: $\lambda = 0.015$ and $\alpha = 1$. For MF and PBMF, we also need to set the number of latent factors. We set it to an initial value of 60, and later perform sensitivity analysis by varying the number of latent factors.

Evaluation Metrics

We adopt three widely used metrics for the evaluation of product recommendation results. The first two metrics are $Precision@k$ and $Recall@k$:

$$Precision@k = \frac{\#correct@k}{k}, \quad Recall@k = \frac{\#correct@k}{\#total\_purchased}.$$

$Precision@k$ and $Recall@k$ reflects the performance of top recommendations. We also use the area under the ROC (Receiver Operating Characteristic) curve, or simply AUC[7], which has been widely used to measure model performance for recommendation tasks. An area of 1 represents a perfect system and an area of 0.5 represents a random system.

Results and Analysis

We present the results of different methods in Table 5 using a different split of training and test set. It can be observed that MP gives the worst results which shows that recommending products simply based on product popularity is not a good strategy for personalized product recommendation. MF outperforms both MP and UCF. Our proposed PBMF method gives superior performance compared to all the baselines. The improvement is more significant with the increasing size of the training set. In particular, we observe an improvement of 12-18% in P@10 and R@10 when the training data are at least doubling the size of the test set.

We also evaluate the performance of matrix factorization based methods, MF and PBMF, by varying the number of latent factors from 0 to 150 with a lap of 30. The AUC results are shown in Figure 3. For easy comparison, we include the UCF results as well. It can be observed that the performance

[7] http://gim.unmc.edu/dxtests/roc3.htm

$\frac{\#training}{\#test}$	Metrics	MP	UCF	MF	PBMF
1:1	P@10	0.006	0.070	0.102	**0.104**
	R@10	0.007	0.049	0.069	**0.072**
	AUC	0.510	0.643	0.685	**0.695**
2:1	P@10	0.017	0.113	0.102	**0.120**
	R@10	0.005	0.033	0.030	**0.034**
	AUC	0.648	0.755	0.767	**0.773**
4:1	P@10	0.042	0.105	0.111	**0.122**
	R@10	0.008	0.021	0.023	**0.026**
	AUC	0.737	0.799	0.814	**0.825**

Table 5: Performance comparisons of different methods. The improvement of PBMF over the other baselines is significant at the confidence level of 0.9.

of both MF and PBMF improves with the increasing number of latent factors. PBMF consistently outperforms MF. In particular, the performance improvement of PBMF over MF is more significant when the number of latent factors is small (< 60). Since it is much more computationally demanding to train models with larger number of latent factors, the above results show that PBMF is indeed desirable in task where efficiency is an important factor to consider.

Conclusion

In this paper, we have made a first attempt to mine product adopter information from reviews and use it for product recommendation in a large dataset. Using the automatically derived product adopter extraction patterns by a bootstrapping-based method, we have found that more than 10% of the reviews contain at least one adopter mention. The adopter mentions are grouped into six categories by using the idea of demographic segmentation in marketing research. Based on the six adopter categories, we propose several distributions to characterize users and products, which are further incorporated into a weighted regularised matrix factorisation (MF) approach. The proposed approach has been shown to generate better recommendation results than the MF approach without considering product adopters' demographic attributes.

We have not considered sentiment analysis in our current work. Product adopter mentions in negative reviews should be used as negative evidence for the recommendation task.

In the future, we plan to combine sentiment analysis with demographic attributes inference from review text to build a more accurate recommendation model. We will also consider grouping adopter mentions into more fine-grained demographic categories and explore other probabilistic methods for product recommendation.

Acknowledgements

Jinpeng Wang and Xiaoming Li were supported by the National Key Basic Research Program (973 Program) of China under grant No. 2014CB340400 and NSFC Grant 61272340. Xin Zhao was partially supported by the Fundamental Research Funds for the Central Universities, the Research Funds of Renmin University of China, and the pilot project under Baidu open cloud service platform. Yulan He was partially supported by Innovate UK under the project number 101779.

References

Adomavicius, G., and Tuzhilin, A. 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering* 17(6).

Cohen, W. A. 2005. *The Marketing Plan*. Wiley.

Ganu, G.; Elhadad, N.; and Marian, A. 2009. Beyond the stars: Improving rating predictions using review text content. In *Proceedings of the 12th International Workshop on the Web and Databases (WebDB)*.

Ganu, G.; Kakodkar, Y.; and Marian, A. 2013. Improving the quality of predictions using textual information in online user reviews. *Information Systems* 38(1):1–15.

Giering, M. 2008. Retail sales prediction and item recommendations using customer demographics at store level. *ACM SIGKDD Explorations Newsletter* 10(2).

Hu, M., and Liu, B. 2004. Mining and summarizing customer reviews. In *Proceedings of the 10th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 168–177.

Hu, Y.; Koren, Y.; and Volinsky, C. 2008. Collaborative filtering for implicit feedback datasets. In *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM)*, 263–272.

Jamali, M., and Ester, M. 2009. Trustwalker: a random walk model for combining trust-based and item-based recommendation. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD)*, 397–406.

Koren, Y.; Bell, R.; and Volinsky, C. 2009. Matrix factorization techniques for recommender systems. *Computer* 42(8):30–37.

Korfiatis, N., and Poulos, M. 2013. Using online consumer reviews as a source for demographic recommendations: A case study using online travel reviews. *Expert Systems and Applications* 40(14).

Lekakos, G., and Giaglis, G. M. 2007. A hybrid approach for improving predictive accuracy of collaborative filtering algorithms. *User Modeling and User-Adapted Interaction* 17(1-2):5–40.

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

Liu, Y.; Huang, J.; An, A.; and Yu, X. 2007. ARSA: A sentiment-aware model for predicting sales performance using blogs. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR)*.

Ma, H.; Zhou, T. C.; Lyu, M. R.; and King, I. 2011. Improving recommender systems by incorporating social contextual information. *ACM Transactions on Information Systems* 29(2):9:1–9:23.

Massa, P., and Avesani, P. 2007. Trust-aware recommender systems. In *Proceedings of the 1st ACM conference on Recommender systems (RecSys)*, 17–24.

McGlohon, M.; Glance, N. S.; and Reiter, Z. 2010. Star quality: Aggregating reviews to rank products and merchants. In *Proceedings of the 4th Int'l AAAI Conference on Weblogs and Social Media (ICWSM)*. The AAAI Press.

Pang, B., and Lee, L. 2008. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval* 2(1-2):1–135.

Pazzani, M. J. 1999. A framework for collaborative, content-based and demographic filtering. *Artificial Intelligence Review* 13(5-6):393–408.

Qiu, L., and Benbasat, I. 2010. A study of demographic embodiments of product recommendation agents in electronic commerce. *International Journal of Human-Computer Studies* 68(10):669–688.

Sarwar, B.; Karypis, G.; Konstan, J.; and Riedl, J. 2001. Item-based collaborative filtering recommendation algorithms. *Proceedings of the 10th international conference on World Wide Web (WWW)*, 285-295.

Seroussi, Y.; Bohnert, F.; and Zukerman, I. 2011. Personalised rating prediction for new users using latent factor models. In *Proceedings of the 22nd ACM Conference on Hypertext and Hypermedia (Hyertext)*, 47–56.

Symeonidis, P.; Tiakas, E.; and Manolopoulos, Y. 2011. Product recommendation and rating prediction based on multi-modal social networks. *Proceedings of the fifth ACM conference on Recommender systems (RecSys)*, 61-68.

Tang, J.; Gao, H.; Liu, H.; and Das Sarma, A. 2012. etrust: Understanding trust evolution in an online world. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 253–261.