Silence is Also Evidence: Interpreting Dwell Time for Recommendation from Psychological Perspective

Peifeng Yin[†] Ping Luo[§] Wang-Chien Lee[‡] Min Wang[♯]

[†,‡] Department of Computer Science & Engineering, Pennsylvania State University
[§] Hewlett Packard Labs China, Beijing, China
[♯] Google Research, Mountain View, CA
{[†]pzy102, [‡]wlee}@cse.psu.edu, [§]ping.luo@hp.com, [♯]minwang@google.com

ABSTRACT

Social media is a platform for people to share and vote content. From the analysis of the social media data we found that users are quite inactive in rating/voting. For example, a user on average only votes 2 out of 100 accessed items. Traditional recommendation methods are mostly based on users' votes and thus can not cope with this situation. Based on the observation that the dwell time on an item may reflect the opinion of a user, we aim to enrich the user-vote matrix by converting the dwell time on items into users' "pseudo votes" and then help improve recommendation performance. However, it is challenging to correctly interpret the dwell time since many subjective human factors, e.g. user expectation, sensitivity to various item qualities, reading speed, are involved into the casual behavior of online reading. In psychology, it is assumed that people have choice threshold in decision making. The time spent on making decision reflects the decision maker's threshold. This idea inspires us to develop a View-Voting model, which can estimate how much the user likes the viewed item according to her dwell time, and thus make recommendations even if there is no voting data available. Finally, our experimental evaluation shows that the traditional rate-based recommendation's performance is greatly improved with the support of VV model.

Categories and Subject Descriptors

H.1.2 [**User/Machine Systems**]: Human factors

Keywords

dwell time, recommendation, psychological

1. INTRODUCTION

Social media provides a good platform for their users to easily publish, view and rate/vote multi-form contents via their computers and smart phones. In such a system a content recommendation service is an important application for both common users and website providers. For the users, popular social media has an overwhelming volume of content with varied quality, making users difficult to distill the information of their interests. Thus, a recommendation service that helps users to find interesting content is highly desirable. On the other hand, for some social web sites (e.g., Amazon[1]), recommendation is often exploited by the providers to encourage more transactions from users, e.g., recommending laptop accessories after a person orders a notebook.

Collaborative filtering (CF) is a widely adopted recommendation technique. CF is mainly based on users' behaviors of voting, rating or buying, which express their opinions on certain items. While there exist various actions, without loss of generality, here we consider only voting for simplicity. CF assumes that users with similar interests exhibit similar voting behaviors and thus measures the similarity of two users based on their historical item voting. If a user has few ratings, the derived similarity would not provide a good basis to make effective recommendation.

Although lots of methods are proposed to alleviate the impact of sparse data in certain degree [28, 26, 3, 7, 24, 5], they need users' explicit opinions to make recommendations. Unfortunately, a common situation is that users often stay quiet instead of taking actions to vote. Based on our analysis of real data collected from a joke sharing mobile application JokeBox[2], a person on average only votes 2 out of 100 accessed items. Therefore, the data sparsity problem is partially a *voting sparsity problem*. We call the phenomenon that a user silently viewing an item without expressing their opinions (i.e., giving a vote) *"silent viewing behavior"*. In this work, we aim to exploit it for recommendation.

To view an item, no matter a vote is eventually given or not, a person has to spend some time on it. The time spent provides valuable information about her interest in the item to some degree. Consider a shopping scenario where a lady was comparing two handbags. Suppose she spent 1 minute on one but 5 minutes on the other. Even if she eventually bought neither one, the time difference indicates her preference to the second handbag. Thus, from the standpoint of a

KDD'13, August 11–14, 2013, Chicago, Illinois, USA.

[1]http://www.amazon.com/
[2]http://itunes.apple.com/us/app/all-in-1-joke-box-no-ads!/id363494433?mt=8

merchant, it may be a good idea to recommend a handbag similar to the second one with a lower price. As these silent viewing behaviors actually indicate their potential interests on certain items, we aim to exploit such information for recommendation. Specifically, we carefully model the *dwell time*, the time a user spends on an item, and convert them into users' "pseudo votes" on items. These pseudo votes are then used to enrich the sparse user-vote matrix and hopefully improve the recommendation performance.

There exists some recent research works in psychology which focus on the dwell time in the recognition process known as *information accumulation* [27]. In their test of *two alternative forced-choice (2-AFC)* task, the participants are required to choose an answer out of two choices (e.g., yes or no). The psychologists proposed a diffusion model to simulate this recognition process, in which people collect evidence for decision making. It assumes that a person has an *action bound* in making choices and would not make decision until the evidence of one choice exceeds the bound. The dwell time reflects the process of information accumulation. Generally, the easier the task is, the shorter the dwell time will be. However, this model can not be applied directly to our case due to the following reasons.

• Experiments of the diffusion model are designed to require participants to follow certain strict rules and then make an explicit response for each testing task. In contrast, viewing items in social web sites is such a casual behavior that people may terminate the viewing process at any time. Neither do they need to give a response.

• Tasks tested with the diffusion model are very specific and usually quite simple (e.g., read a word, scan a short symbol string). The dwell time is easy to model in those tasks. In our case, however, the dwell time on items may vary a lot due to the factors from both items and persons. On one side items may differ not only in their form and volume (e.g., different length of articles, video, different size of pictures, etc.), but also in their quality to attract further consumption. On the other side there are many subjective human factors to affect the dwell time. For example, different people receive information at different speed and the time of consuming the same item (e.g., reading an article) may differ from person to person. Furthermore, some people are very "picky" and are willing to spend time only on items which match their tastes and expectations. However, some people are rather tolerant and would like to read all items despite their diversified quality. All these factors on the sides of items and persons jointly determine the dwell time of a person on a certain item.

• In 2-AFC, positive correlation is observed between dwell time and answer accuracy. Thus the action bound is subjectively set by the researchers in accordance with the participant's answer accuracy, where higher accuracy indicates a more careful personality and thus a higher action bound. In real life where people view and vote items, there are no such "right" or "wrong" answers as in 2-AFC, and thus the bound, if existing, can not be determined heuristically.

In this work, we propose a *Viewing-Voting (VV)* model to capture both the silent viewing and explicit voting behaviors, and explain the implication behind a person's dwell time on an item. We assume that each item has a quality value and each person has multiple *latent action bounds (LABs)*. These LABs determine the expectation levels of items that may motivate the person's viewing and voting behavior. When a person begins to view an item, one of her LABs is selected. Then, we have the following three situations. First, if the item quality is lower than the bound, the dwell time tends to be short, suggesting that the user does not enjoy the article and thus stops reading before reaching the end. Second, if the quality matches the bound, the dwell time is close to the time needed to "comprehend the item" (i.e., finish reading the article). Finally, if the quality is much higher than the bound, the dwell time is long because the person tends to read a story more before letting it go, indicating that she finds it really good. This models the viewing behavior of users.

As for the modeling of the voting behavior, we consider that if the selected bound is smaller than the item quality, the user is more likely to give a positive vote. Otherwise, the user may simply keep silent. With the VV model, even if a user leaves no vote after viewing an item, we can still estimate the user's possible opinion based on the dwell time and exploit it for recommendation. To sum up, our contribution is three-fold.

• By analyzing the data, we discover users' infrequent voting behavior and coin the voting sparsity problem. We argue that this problem can be addressed in content viewing applications by mining the dell time.

• We propose a Viewing-Voting (VV) model to i) explain people's silent viewing behavior based on their dwell time on the item; and ii) model the users' voting behavior. Based on the VV model, we develop a strategy that interprets the dwell time to user's possible vote, referred to as *pseudo vote* and exploit it for recommendation.

• We conduct extensive experiments on a real dataset and demonstrate the improvement of conventional recommendation techniques when combined with VV model.

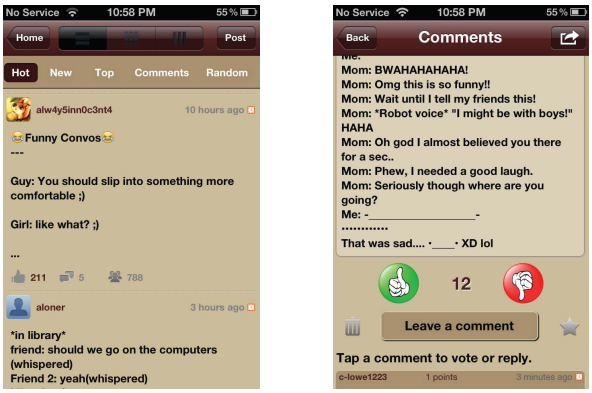
The rest of the paper is organized as follows. Section 2 presents our data analysis results, which serve as a guidance for the model development. Section 3 and Section 4 provide details of the VV model and show how it interprets dwell time for recommendation. Section 5 and Section 6 respectively shows the evaluation result and reviews the related work. Finally Section 7 concludes the paper.

2. PRELIMINARIES

In this section we perform a data analysis and show the results, some of which guide the design of the VV model. The analysis mainly consists of three parts, (i) revealing the fact of vote sparsity, (ii) testing the existence of the action bound, and (iii) exploring the characteristics of dwell time.

2.1 Data Analysis

The data we use are the log of JokeBox from June to November, 2011. JokeBox is a popular iPhone application, where people publish and vote jokes. Figure 1 shows the snapshots of its main interfaces. When the application starts, a list of jokes with abstract is displayed as shown in Figure 1a. Users need to tap on some joke to see the full content, whose interface is shown in Figure 1b. After reading a joke, user can vote for (i.e., the green hand in Figure 1b) or against (which is the red hand in Figure 1b) the joke or simply retreat to the list by tapping the "return" button at the upper-left corner. The log records such actions as "tap", "vote" and "retreat" with time stamp. The dwell time is obtained by computing the time stamp difference between "tap" and "retreat" of the same user on the particular



(a) JokeBox Snapshot 1 (b) JokeBox Snapshot 2

Figure 1: Snapshot of JokeBox.

item. There are in total 638,899 records containing 108,743 users and 143,258 items (jokes)[3]. The log also records the final vote situation for each item, i.e., the number of positive votes minus that of negative votes, which, as an example, is 12 for the joke in Figure 1b. This value to some degree reflects the public evaluation for this item. Thus, in this work we treat it as the *quality* of the item. Additionally, the statistics from our preliminary analysis show that the negative vote occupies a rather small proportion (1,395 negative votes in 143,258 items) and thus in this work we only focus on analyzing and modeling positive voting and silent viewing behavior.

2.1.1 Cause of data sparsity

We first find statistics on the frequency of user giving vote and define a metric *user-voting ratio* r_u in Equation (1).

$$r_u = \frac{\text{Number of votes given by } u}{\text{Number of items viewed by } u} \times 100\% \qquad (1)$$

The statistics of r_u is shown in Table 1. We can see that most (97.91%) of users' voting ratio is quite small (less than 30%). Moreover, 95.93% of people never give a vote. A further analysis shows that the average user-voting ratio is only 2.02%. Thus the data sparsity problem, or exactly the *voting sparsity problem*, is caused by the user's infrequent voting behavior although they may be quite active in viewing items.

Table 1: Statistics of user-voting ratio

r_u(%)	No.	Percent (%)	Cumu. percent (%)
0	90202	95.93	95.93
0∼30	1861	1.98	97.91
30∼60	1067	1.13	99.04
60∼100	903	0.96	100

2.1.2 Existence of action bounds

We then test i) whether there exists action bounds for users when viewing and voting an item and ii) whether the bound is different from person to person.

We firstly group people with regarding to the number of positive votes they give. Intuitively a person's bound, if existing, would be high if she had high expectation on the item quality and hardly gave a positive vote. Next, for each person in the group, we identify all her accessed items and classify them into two categories, "like" and "neutral", depending on whether she gave (positive) vote to the items or not. Then, for each person we compute the average quality values of her accessed items from the two categories, respectively. Finally, we average the quality values (i.e., as mentioned above, the final vote status) of these two categories over all persons with the same number of positive votes. The result is shown in Figure 2. Note that as the number of positive vote increases, the sample size of users who give exactly that number of votes decreases. Therefore we only display the result of groups that have more than 20 samples in Figure 2.

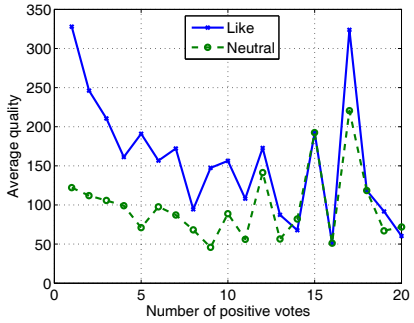


Figure 2: Existence of action bound

It can be easily seen that for the first 10 groups the "like" category (blue solid line) is above the "neutral" one (green dashed line). It indicates that on average the quality of the items in the "like" category for a person is bigger than the quality of her accessed items that failed to receive her positive vote. Also, the average item qualities for both categories have a decreasing trend as the number of positive vote in x axis increases. This supports the action bound hypothesis. The persons who give less positive votes may have higher level of expectation on the item quality, namely the higher action bound. Thus, only the items with higher quality, which exceeds her action bound, can motivate her to vote positively. Finally, for large numbers of positive vote ($x > 10$), the trend of these two categories becomes unclear. One possible reason is that people may have multiple bounds whose values are quite different. As the number of positive votes increases, different bounds are likely to be selected to guide the voting behavior and thus affect the statistic curve.

2.1.3 Trend of dwell time

Next we explore the trend of the dwell time. As mentioned in Section 1, the dwell time for item viewing is different from item to item due to their variant formats (e.g., text, picture, video). For jokes which consist of text, the intuition is that it should be proportional to the text length. We group all the viewing events according to the item length. For the viewing events with the similar item length we further classify them into two categories, those that end with a positive vote from users (denoted as "like") and those with silent viewing (denoted as "neutral"). For each category we calculate their average dwell time and plot it with regarding to the item length in Figure 3.

From the figure, we can see that for both categories the dwell time is positively proportional to the item length. This

[3]Not all users' actions were recorded due to the version issue. However, since the recorded users are quite random, the analysis result is still reliable.

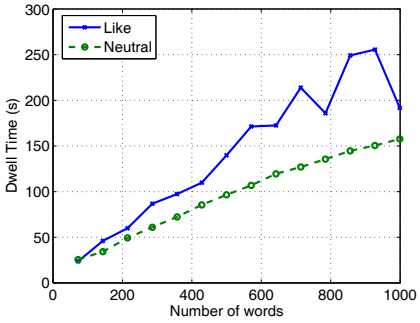
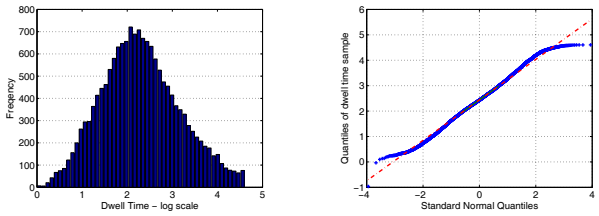


Figure 3: Trend of dwell time w.r.t item length

is reasonable because longer texts take more time for people to read. Another interesting observation is that the blue solid line is always above the green dashed line, suggesting that the average dwell time for positively voted items is longer than that of the neutral ones when the item length is similar. Recall that in Figure 2 we found positively voted item has a higher quality than neutral. This suggests that people tend to spend more time on the items of higher quality. This observation makes sense because in reality when a person meets a good item, e.g., a really funny joke, she is more likely to read it multiple times, resulting in longer dwell time.

Finally we explore the distribution of dwell time. As shown in Figure 3, the average dwell time correlates with the story's length. Therefore we first group all stories according to their length. Then we select the group of items with the same length which has most number of views and count the frequency of the dwell time. Figure 4a shows the frequency for stories whose length ranges from 36 to 46. Note that the x-axis is in log-scale. The statistical result indicates that the dwell time may satisfy a log-Gaussian distribution. To further prove this hypothesis, we compare the log value of dwell time sample to the standard Gaussian distribution and plot the quantile-quantile curve in Figure 4b. As can be seen, the blue markers closely lie on the red straight line, indicating that the two distributions are linearly related. Therefore it is proper to use a log-Gaussian distribution to model the dwell time.



(a) Raw frequency (b) Q-Q plot

Figure 4: Characteristics of dwell time

In the end of this section, we summarize our observations into principles that will guide the design of our model. Firstly, people tend to have multiple action bounds which differ from person to person (see Figure 2). Secondly, the length of a textual item determines the expected dwell time, which at the mean time is affected by the difference between the item's quality and the user's action bound (see Figure 3). Finally, it is proper to choose log-Gaussian distribution to model the dwell time (see Figure 4).

3. VIEWING-VOTING MODEL

Table 2: List of Symbols

Symbol	Meaning
u	the user
b	latent action bound (LAB)
π	probability of LAB selection
r	reading speed
α	quality sensitivity
v	vote
t	dwell time
q	the item quality
l	the item length
p	the Bernoulli parameter

In this section we discuss the details of the model. Table 2 lists the symbols we used in the work. We assume that each user has several latent action bounds (LABs). When viewing an item, the user will randomly select a LAB, which, together with the quality of the item, jointly affects the user's voting behavior, i.e., to leave a vote or not. Generally, if the selected LAB is bigger than quality, the user is less likely to leave a vote and vice versa. The LABs can be treated as the expectation of the person to the quality of the item. The higher the user's expectation is, the harder the item can entertain her, and vice versa. However, different from item's quality that can either be measured or reflected (by people's votes), the "expectation" is hard to measure. Therefore we model it as a hidden parameter.

Besides the voting, another value, the dwell time, which is corresponding to the viewing behavior, is also generated. This is a metric that measures how long the user would stay on this item before moving forward. In general, it is affected by LAB and quality in a way that is similar to how the vote is produced. For big LAB and small quality, the dwell time tends to be short since the item can hardly entertain the user. For small LAB and big quality, the dwell time would be long as the user is extremely attracted to the item.

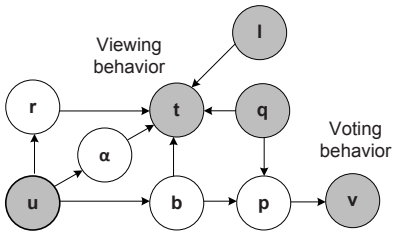


Figure 5: The graphical model. The shaded circles stand for observable values while the unshaded ones for latent variables.

The graphical model is shown in Figure 5. Each user u has a set of LABs. Each item consists of two components, i.e., property l (in our case it is the length of the joke) and quality q. The quality q and bound b jointly generate the vote v, where 1 means positive vote and 0 no vote.

The dwell time t is determined by two groups of factors, (i) user-related and (ii) item-related. As the name suggests, the former one stands for the personalized factors that affect the dwell time. The latter one, on the other hand, may be different as the item differs. Specifically in the graph model, the t is generated by five parameters, among which LAB b, *reading speed* r and *quality sensitivity* α are user-related

while the other two, namely q and l are item-related. The speed r measures how quickly a person can read while the sensitivity α reflects how sensitive the person is to items of diversified quality. The details will be discussed later in Section 3.2.

The item-related factors have different semantics with regarding to the different item types. In this work, we focus our attention on the textual item and discuss these two factors in the context of an article. In the graphical model, the q represents the quality of the item and l represents the length of the article.

Algorithm 1 Generation process

1. Choose a LAB $b \sim \texttt{Multinomial}(\pi_i)$, $1 \le i \le k$
2. Choose a parameter $p \sim \texttt{Be}(q, b)$
3. Choose a vote $v \sim \texttt{Bernoulli}(p)$
4. Choose a time $\log t \sim \mathcal{N}(\mu, \sigma^2)$

The generation process is summarized in Algorithm 1. The user first selects one of k LABs, where the probability of selecting i^{th} bound b_i is π_i. After the LAB b is generated, it goes along with the item's quality q to determine the vote, which is modeled as a Bernoulli process and the parameter p satisfies a Beta distribution with parameters (q, b). We will discuss its details in Section 3.1.

Finally, a dwell time t is generated by a log-Gaussian distribution whose mean is μ and variance is σ^2, as suggested in Figure 4. Details of the dwell time model as well as these parameters are discussed in Section 3.2.

3.1 Model of Voting Behavior

Here we discuss the model of voting. As described earlier, it is a Bernoulli process, where the probability of positive vote is p while the neutral is $1 - p$. The parameter p is produced by a Beta distribution $\texttt{Be}(q, b)$ defined in Equation (2).

$$\texttt{Be}(p; q, b) = \frac{\Gamma(q+b)}{\Gamma(q)\Gamma(b)} p^{q-1}(1-p)^{b-1} = \frac{p^{q-1}(1-p)^{b-1}}{\texttt{B}(q, b)} \quad (2)$$

where q is the item's quality and b is the user's LAB.

In probability theory, Beta distribution is often used to describe a prior distribution of a parameter for some distribution, e.g., Bernoulli distribution. Specifically, the q and b jointly determine the probability distribution of p. When $q > b \ge 1$, the value of p is more likely to be large. In case of $1 \le q < b$, the generated p is closer to 0. That means, the user is more likely to give a positive vote if the quality of the item is bigger than the LAB b. If not, a silent viewing may possibly happen.

To sum up, given a selected LAB b and an item whose quality is q, the probability of a positive vote is given in Equation (3).

$$\begin{aligned} P(v=1|b,q) &= \int_0^1 P(v=1, p|b, q)dp \\ &= \int_0^1 p \cdot \texttt{Be}(p; q, b)dp = \frac{q}{q+b} \end{aligned} \quad (3)$$

Similarly, the probability of no vote ($v = 0$) is shown below:

$$P(v=0|b,q) = 1 - P(v=1|b,q) = 1 - \frac{q}{q+b} = \frac{b}{q+b} \quad (4)$$

Finally, Equation (3) and (4) can be unified as below.

$$P(v|b,q) = \frac{(1-v)\cdot b + v \cdot q}{q+b} \quad (5)$$

3.2 Model of Dwell Time

In this section we discuss the model of dwell time. Specifically, we consider the following three cases with regarding to selected LAB b and item quality q:

- $b > q$. The item is not good enough to motivate the user to vote. In this case, the user would not spend too much time on this item and she would even quit before reaching the end of the passage.
- $b \approx q$. The item's quality is very close to the user's LAB, which may lead to the user's hesitance of voting. Therefore, the user may continue to read the item to its end. And the time is related to the item's property, i.e., the length of the article.
- $b < q$. The item's quality goes beyond the user's expectation and naturally the user may spend much more time reading this item and may remain on it after finishing reading, making the time longer than expected.

Based on the result of statistical analysis obtained in Section 2, we use log-Gaussian distribution to model the dwell time. The mean value μ is determined by the item's length l, quality q, the selected LAB b as well as two personalized parameters *reading speed* r and *quality sensitivity* α. Recall in Figure 3 we found that the average dwell time is linearly correlated to the item's length and thus l/r measures how long the person would spend reading a whole article of length l. The sensitivity $\alpha > 0$ determines the influence of the difference between quality and LAB, i.e., $q - b$ on the dwell time. Given the item's length l, quality q and the user's LAB b as well as r and α, the probability that the user would spend time t on this item is given in Equation (6).

$$\begin{aligned} &P(t|l, q, b, r, \alpha) = \mathcal{N}(\log t; \mu(r, l, \alpha, q, b), \sigma^2) \\ &= \frac{1}{\sqrt{2\pi\sigma^2}} \cdot e^{-\frac{(\log t - \mu)^2}{2\sigma^2}} \\ &\text{where } \mu(r, l, \alpha, q, b) = \log(l/r) + \alpha(q - b) \end{aligned} \quad (6)$$

Note that if $q > b$, the mean dwell time would increase and vice versa. The sensitivity α determines the time growth-decay proportion and is different from person to person. Big α may indicate a "picky" person whose dwell time is heavily quality-dependent, i.e., spending little time on trash but much time on high-quality items. The small α, on the other hand, suggests a "tolerant" person whose dwell time does not vary too much on items of diversified quality.

3.3 Model Learning

Based on earlier discussion, the model parameters are LAB distribution $\pi_i, 1 \le i \le k$, the series of LABs b_i, reading speed r, quality sensitivity α and dwell time variance σ^2. Let θ denote all of these parameters. The learning is a process to determine proper values for θ in order to maximize the probability of observed data, i.e., the item's quality q and length l, the user's vote $v \in \{0, 1\}$ as well as her dwell time t.

Let $(\mathbf{Q} = \{q_1, \cdots, q_n\}, \mathbf{L} = \{l_1, \cdots, l_n\})$ denote the n items accessed by the user while $\mathbf{V} = \{v_1, \cdots, v_n\}$ and $\mathbf{T} = \{t_1, \cdots, t_n\}$ respectively stand for the vote and dwell time of the user on the corresponding item. Finally $\mathbf{B}$ are the

unobserved LABs that affect the user's behavior when viewing the item. A loss function with constraint $\sum_{i=1}^{k}\pi_i = 1$ is defined in Equation (7).

$$\begin{aligned}\mathfrak{L}(\theta;\mathbf{Q},\mathbf{L},\mathbf{V},\mathbf{T},\mathbf{B}) &= \log P(\mathbf{V},\mathbf{T},\mathbf{B}|\mathbf{Q},\mathbf{L},\theta) + \lambda(\sum_{i=1}^{k}\pi_i - 1)\\ &= \sum_{i=1}^{n}\sum_{j=1}^{k}\tau_{ij}\log P(v_i,t_i,b_j|q_i,l_i,\theta) + \lambda(\sum_{i=1}^{k}\pi_i - 1)\\ &= \sum_{i=1}^{n}\sum_{j=1}^{k}\tau_{ij}(\log P(v_i|q_i,\theta_j) + \log P(t_i|l_i,q_i,\theta_j) + \log\pi_j)\\ &\quad + \lambda(\sum_{i=1}^{k}\pi_i - 1)\end{aligned} \tag{7}$$

where λ is a Lagrange multiplier and τ_{ij} is a judge function defined in Equation (8).

$$\tau_{ij} = \begin{cases}1; & \text{if LAB } b_j \text{ is selected for item } i\\ 0; & \text{otherwise}\end{cases} \tag{8}$$

We use EM (expectation-maximum) algorithm [8] to solve the problem. In the following, θ^s denotes the parameter values for the s^{th} iteration.

E-step.

$$\begin{aligned}E^{s+1}(\tau_{ij}) &= P(b_j^s|v_i,t_i,q_i,l_i,r^s,\alpha^s)\\ &= \frac{P(v_i|b_j^s,q_i)P(t_i|l_i,q_i,b_j^s,r^s,\alpha^s)P(b_j^s)}{\sum_{m=1}^{k}P(v_i|b_m^s,q_i)P(t_i|l_i,q_i,b_m^s,r^s,\alpha^s)P(b_m^s)}\\ &= \frac{P(v_i|b_j^s,q_i)P(t_i|l_i,q_i,b_j^s,r^s,\alpha^s)\pi_j^{\alpha}}{\sum_{m=1}^{k}P(v_i|b_m^s,q_i)P(t_i|l_i,q_i,b_m^s,r^s,\alpha^s)\pi_m^{\alpha}}\end{aligned} \tag{9}$$

The definition of $P(v_i|b_j,q_i)$ and $P(t_i|l_i,q_i,b_j,r,\alpha)$ are in Equation (5) and (6).

M-step.

$$\pi_j^{s+1} = \frac{\sum_{i=1}^{n}E^{s+1}(\tau_{ij})}{\sum_{i=1}^{n}\sum_{j=1}^{k}E^{s+1}(\tau_{ij})} = \frac{\sum_{i=1}^{n}E^{s+1}(\tau_{ij})}{n} \tag{10}$$

$$\begin{aligned}&\sum_{i=1}^{n}E^{s+1}(\tau_{ij})\left(\frac{1-v_i}{(1-v_i)b_j^{s+1}+v_iq_i} - \frac{1}{b_j^{s+1}+q_i}\right)\\ &-\sum_{i=1}^{n}E^{s+1}(\tau_{ij})\left(\frac{\alpha^s(\log t_i - \log(l_i/r^s) + \alpha^s(b_j^{s+1}-q_i))}{(\sigma^s)^2}\right) = 0\end{aligned} \tag{11}$$

$$\begin{aligned}\sigma^{s+1} &= \sqrt{\frac{\sum_{i=1}^{n}\sum_{j=1}^{k}E^{s+1}(\tau_{ij})(\log t_i - \mu_{ij}^s)^2}{\sum_{i=1}^{n}\sum_{j=1}^{k}E^{s+1}(\tau_{ij})}}\\ &= \sqrt{\frac{\sum_{i=1}^{n}\sum_{j=1}^{k}E^{s+1}(\tau_{ij})(\log t_i - \mu_{ij}^s)^2}{n}}\end{aligned} \tag{12}$$

$$\log r^{s+1} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{k}E^{s+1}(\tau_{ij})(\log l_i - \log t_i - \alpha^s(b_j^s - q_i))}{\sum_{i=1}^{n}\sum_{j=1}^{k}E^{s+1}(\tau_{ij})} \tag{13}$$

$$\alpha^{s+1} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{k}E^{s+1}(\tau_{ij})(\log(l_i/r^s) - \log t_i)(b_j^s - q_i)}{\sum_{i=1}^{n}\sum_{j=1}^{k}E^{s+1}(\tau_{ij})(b_j^s - q_i)^2} \tag{14}$$

All parameters except b_j can be directly computed while b_j can be obtained by finding the root for Equation (11) with Newton's method.

4. RECOMMENDATION

Here we introduce how to use the proposed model to interpret the silent viewing behavior for recommendation. Specifically, we split the task into three phases. In phase I, given a user's dwell time on an item, the model is applied to find her LAB that is most likely to be selected to guide the viewing behavior, referred to as *LAB choice estimation*. In phase II, namely *vote prediction*, the probability is computed that how likely the user would vote for the item given this estimated LAB. These "pseudo votes" are used to enrich the original sparse user-vote matrix. Finally in phase III, conventional rating-based recommendation techniques, e.g., CF, is applied to make recommendation based on both the actual and pseudo votes.

Phase I.

Formally, for a particular item o, let q^o and l^o respectively denote its quality and length. Suppose the dwell time of i^{th} user on the item is t_i^o. Let $u_i = \{\langle\pi_{i_j}, b_{i_j}, \sigma_i, r_i, \alpha_i\rangle | 1 \le j \le k\}$ denote the i^{th} user's profile, which is obtained after model training. LAB choice estimation is to find a LAB $b_{i_o}^*$ that is most likely to be selected to generate the observed dwell time, as shown in Equation (15).

$$\begin{aligned}b_{i_o}^* &= \text{argmax}_{b_{i_j}} P(t_i^o|\pi_{i_j}, b_{i_j}, \sigma_i, r_i, \alpha_i)\\ &= \text{argmax}_{b_{i_j}} \pi_{i_j}\mathcal{N}(\log t_i^o; \mu(r_i, l^o, \alpha, q^o, b_{i_j}), \sigma_i^2)\end{aligned} \tag{15}$$

Phase II.

After a proper estimated LAB $b_{i_o}^*$ is found, we can interpret the user u_i's dwell time to her possible preference on that item, which is modeled as the expected vote as given in Equation (16).

$$E(v_i^o|q^o, b_{i_o}^*) = 1\cdot\frac{q^o}{q^o + b_{i_o}^*} + 0\cdot\frac{b_{i_o}^*}{q^o + b_{i_o}^*} = \frac{q^o}{q^o + b_{i_o}^*} \tag{16}$$

Phase III.

After the first two phases, the "pseudo votes" from the VV model enrich the original sparse voting matrix, on which conventional recommendation techniques can be applied.

5. EVALUATION

For evaluation we use the real log from JokeBox as introduced in Section 2. Although using multiple data would strengthen the work, it is really hard to obtain such view-vote data from another domain. In our experiments, we only keep the users who have viewed no less than 20 items. As a result, we obtain 960 users and 19,196 items, among which there are 2,053 votes and 33,158 silent views. As the default in our experimental setting, the number of LABs is set to 5 unless noted explicitly.

For evaluation, we split the vote data into four parts and conduct 4-fold cross-validation. For every round, three parts of the data are used as the training data to obtain user profiles with Equation (10) to Equation (14). Then the remaining part is used as test data.

Two metrics are exploited in the evaluation. The first one is referred to as *hit ratio*. For each target user, a list of candidate items are ranked and recommended. Among them some do receive positive vote from the target user, treated

as a "hit". Note that a random recommendation could also rank these candidate items by randomly generating a recommendation list. The hit ratio measures the improvement of the proposed recommendation solution over the random recommendation. The performance of a random recommendation, as shown in [33], can be represented as $\frac{|M|}{|C|\cdot|U|}$ where C is the collection of candidate items, U is the collection of users while M is the set of test data. Formally, the cut-off hit ratio at rank N is defined in Equation (17).

$$hratio@N = \frac{\text{Number of hits before } N}{N} \times \frac{|C| \cdot |U|}{|M|} \quad (17)$$

Note that higher hit ratio indicates better performance.

The second adopted metric is *hit rank*. Suppose the ranks of hits are $\langle r_1, \cdots, r_m \rangle$, where $1 \leq r_1 < \cdots < r_m \leq |R|$ and $|R|$ is the size of recommendation list. Ideally, a perfect ranking scheme should rank all items in the head of the list and results in such equation $r_i = i$. The hit rank measures how close the ranking list of the proposed solution is to the ideal one's. Formally, it is computed via dividing the sum of r_i by that of a perfect ranking scheme, which is $\frac{N\cdot(N+1)}{2}$. The formal definition is given in Equation (18).

$$hrank@N = \frac{2\sum_{i=1}^{N} r_i}{N \cdot (N+1)} \quad (18)$$

Since $r_i \geq i$, smaller value of hit rank indicates a better recommendation strategy.

5.1 Model Initialization

As for the parameter initialization, the starting value for impact α_0^u is set to 1. Suppose a user u's viewing history is $\{\langle q_i^u, l_i^u, t_i^u \rangle | 1 \leq i \leq n_u\}$, where q_i^u and l_i^u are the item's quality and length, t_i^u is the user's dwell time spent on her i^{th} accessed item. The user's reading speed r_0^u is initially set as in Equation (19).

$$r_0^u = \frac{\sum_{i=1}^{n_u} l_i^u}{\sum_{i=1}^{n_u} t_i^u} \quad (19)$$

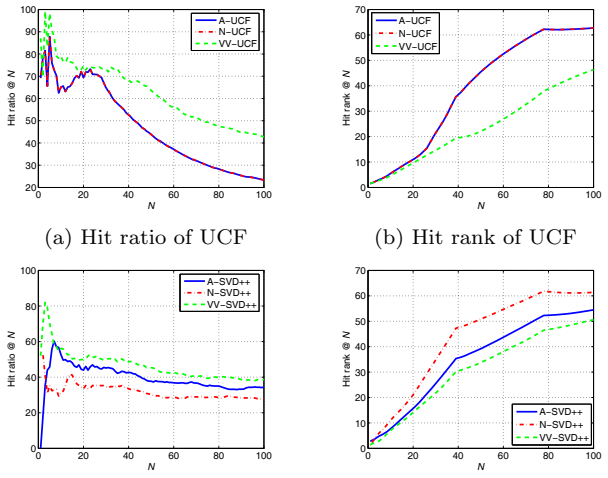
The variance $(\sigma_0^u)^2$ is initially set as the variance of $\log t_i^u$ as in Equation (20).

$$(\sigma_0^u)^2 = \frac{\sum_{i=1}^{n_u} (\log t_i^u - \frac{\sum_{i=1}^{n_u} \log t_i^u}{n_u})^2}{n_u} \quad (20)$$

The probability of LAB selection π_j is initialized as a uniform distribution. For the initial values of LABs, we select random numbers from 1 to 1000 [4].

5.2 Experiment

In this section we show the results of evaluation. We use two variants of collaborative filtering, i.e., user-based collaborative filtering (UCF) [12] and SVD++ [16], both implemented by MyMediaLite [11]. We are aware of that there are many Matrix Factorization variants, e.g., [1, 10] and usually they embed such extra information as user/item meta data, item's textual content. As we believe that our proposal is complementary to existing techniques, in this evaluation we only demonstrate how much improvement our solution brings to the rating-based recommendation techniques that adopts a naive interpretation of silence view. Particularly we adopt two baselines, respectively denoted as Aggressive and Neutral. The former interprets the silent viewing behavior as a rating of 0.1 while the latter treats it as a rating of 0.5 (neutral). Recommendations that adopt these three techniques (Aggressive, Neutral, VV) are denoted as *A-UCF/SVD++*, *N-UCF/SVD++* and *VV-UCF/SVD++*.



(a) Hit ratio of UCF (b) Hit rank of UCF

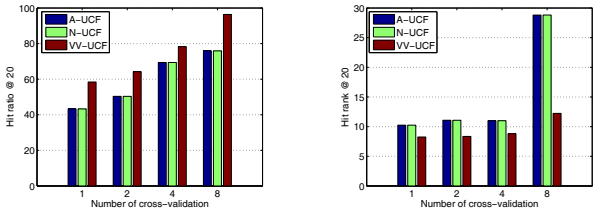
(c) Hit ratio of SVD++ (d) Hit rank of SVD++

Figure 6: General comparison

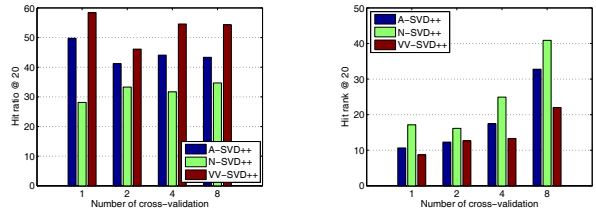
The general performance of the six solutions are shown in Figure 6. We can see that methods that combine with our proposed VV model (VV-UCF/SVD++) benefit an improvement over those with baseline (A/N-UCF/SVD++). Both Aggressive and Neutral scheme adopt a naive way of interpreting dwell time, either to negative vote or neutral. Therefore it is not as accurate as the VV model which provides a complex simulation of the dwell time that involves multiple subjective human factors such as expectation (LAB), reading speed (r) and the sensitivity of quality difference (α). Note that in Figure 6a and Figure 6b, A-UCF and N-UCF display equal performance. For UCF, a cosine similarity metric is adopted to measure the preference similarity between people. Therefore different assigned values, e.g., 1 or 2.5, for the unvoted items does not change the relative similarity between users, leading to the equal performance of A-UCF and N-UCF.

We then test the impact of data sparsity. Note that in previous experiment four-fold cross-validation is used. That means 25% of voting data is used as test data. Here, we evaluate the performance with regarding to different numbers of cross-validation, where bigger value indicates more training data and less test data and vice versa. The performance where $N = 20$ is shown in Figure 7. From Figure 7a, we can see that the increase of cross-validation number brings better recommendation performance for UCF since more training data is available. In Figure 7c however, the data sparsity does not seem to have an obvious impact on SVD++. It may be due to the strength of SVD++ to UCF. Instead of basing on the raw user-item voting data, the SVD++ creates virtual feature space for each user and item, thus may be more resistant to data sparsity than UCF. Finally in Figure 7b and Figure 7d, the performance degrades as the number of cross-validation increases for both UCF and SVD++. Recall the definition of these two metrics in Equation (17)

[4]We tested different values in the experiment and found no significant difference in performance.



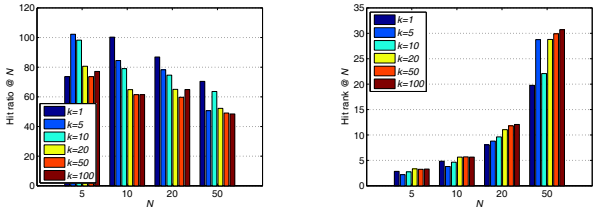
(a) Hit ratio of UCF (b) Hit rank of UCF



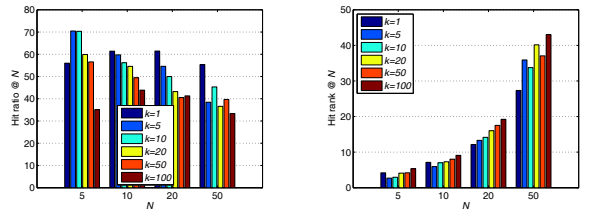
(c) Hit ratio of SVD (d) Hit rank of SVD

Figure 7: Evaluation on the impact of data sparsity

and Equation (18), we can see that compared with hit ratio, hit rank focus more on the ranks of the recommended item. When the training data increases, the test data decreases, leading to more noise in candidate item. However, we can see that VV-SVD++ suffers less degrade compared to baseline, demonstrating its accurate interpretation of dwell time.



(a) Hit ratio of VV-UCF (b) Hit rank of VV-UCF



(c) Hit ratio of VV-SVD++ (d) Hit rank of VV-SVD++

Figure 8: Evaluation on the impact of k

In the final experiment, we test the impact of LAB size *lab*. Figure 8 shows the performance of UCF and SVD++ that are combined with VV model w.r.t different LAB size. As can be seen, generally smaller k achieves better performance. Particularly when $N > 5$ for hit ratio and $N > 10$ for hit rank, VV model with single LAB display best performance. This scenario suggests that for this joke-reading dataset, people are quite consistent in terms of expectation and small number of LAB fits the dataset better than large one. A further check of learned user profiles reveals that when $k > 1$, the learned dwell time variance σ^2 for all users tend to be small. But when $k = 1$, for a small number of people, their learned dwell time variance σ^2 is extremely huge, which means their time spent on items vary a lot and the prediction of their dwell time would be quite inaccurate. This observation provides such a possibility that besides the values of LAB, the number k might be another personalized parameter, which will be explored in future work.

6. RELATED WORK

Our work is related to three areas, i) data sparsity in recommendation, ii) implicit user feedback modeling and iii) diffusion model in psychology.

Data Sparsity in Recommendation. Data sparsity is a really big issue that impacts the performance of collaborative filtering in recommendation. Different approaches have been developed to incorporate various features on users and items into recommendation to tackle this problem. To build the connections among users, new similarity measurement were proposed to find the similar users in [3, 5]. To create new connections among items, different item-based similarity measures are proposed for different types of items, such as textual items [23, 28], POIs (point-of-interest) [32], movies [24] etc. Furthermore, Matrix factorization (MF) [17] is proposed to flexibly embed various features. For instance, in [22, 21], social network is imported to constrain the factorization process in such a way that socially connected people tend to have similar user feature vectors. Also in [1], Agarwal et. al. proposed *fLDA*, a hybrid recommendation method that combines matrix factorization with LDA, aiming to exploit items with rich textual information to improve the rate prediction. To sum up, these works are all based on the condition that users provide explicit opinions. Our work tries to throw away this burden, and tackle the sparsity problem from a new angle, i.e., predicting user's interest in items according to her dwell time.

Study on Implicit Feedbacks. There are some previous works focusing on different types of implicit feedbacks, e.g., "playcount" of music tracks or albums [14], frequency of visits to a content or category [25] and so on. Some methods are also proposed to integrate implicit feedbacks to the rating-based solution [19]. We believe that our model is complement to these works.

There are a few existing works to study dwell time as implicit feedback of users. In [2, 4, 6], dwell time is treated as an extra feature to rank the relevance of retrieved web page to user's query. In [15, 30], the correlation between display time and document usefulness/relevance is studied in information-seeking task. In [20], a system BrowseRank was developed which makes use of user's browsing behavior to rank the importance of web page. Other works [13, 31] try to make use of people's watching time for recommending TV shows or programs. Recently in [25], a positive correlation is observed between display time of picture and user's high ratings. Although Liu et. al. [18] used Weibull distribution to model people's dwell time on web pages, we found that a log-Gaussian distribution is more proper for the dwell time on jokes, as shown in Figure 4. Furthermore, unlike previous works that simply interpret longer dwell time to positive feedback, we explore the latent human factors that determine the dwell time by borrowing the concepts from Psychology, which to our best knowledge is the first work.

Diffusion Model in 2-AFC Task. Psychologist proposed a diffusion model [27] to simulate a person's response time when facing a two-alternative forced choice (2-AFC) task. The participant is required to make a judgement on

a simple task, e.g., whether a given letter string is a legal English word [29]. Researchers measure the response time as well as the answer precision. The diffusion model, also known as *Wiener diffusion process* [29], was first studied in [9] and later implemented by Ratcliff et. al. [27]. It aims to help scientists understand the human recognition process with regarding to factors such as gender, age and so on. The model assumes each person has a response threshold and the choice is made when the accumulated evidence hits the threshold. The response time is also determined by how fast the information is accumulated, namely the drift rate as in [27], which is a normally distributed variable.

As mentioned in Section 1, our scenario (item-viewing in social web) is much more complex than 2-AFC task and the diffusion model can not be directly applied.

7. CONCLUSION AND FUTURE WORK

In this work we propose a Viewing-Voting (VV) model to exploit dwell time for recommendation. Traditional recommendation strategies are based on the opinion-expressing behaviors and do not consider silent viewing behavior. The VV model is developed to bridge the gap by correctly interpreting the dwell time to "pseudo vote". As the experiment shows, the performance of traditional recommendation is greatly improved with the support of our VV model.

As for future work, we will study the trend of dwell time with regarding to different item formats (e.g., audio, video, picture etc), and consider different application scenarios (e.g., online shopping, mobile APP recommendation etc).

8. REFERENCES

[1] D. Agarwal and B.-C. Chen. flda: matrix factorization through latent dirichlet allocation. In *WSDM*, pages 91–100, 2010.

[2] E. Agichtein, E. Brill, and S. Dumais. Improving web search ranking by incorporating user behavior information. In *SIGIR*, pages 19–26, 2006.

[3] H. J. Ahn. A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem. *Inf. Sci.*, 178(1):37–51, 2008.

[4] M. Bilenko and R. W. White. Mining the search trails of surfing crowds: identifying relevant websites from user activity. In *WWW*, pages 51–60, 2008.

[5] J. Bobadilla, F. Ortega, A. Hernando, and J. Bernal. A collaborative filtering approach to mitigate the new user cold start problem. *Knowledge-Based Systems*, 26(1):225–238, 2012.

[6] O. Chapelle and Y. Zhang. A dynamic bayesian network click model for web search ranking. In *WWW*, pages 1–10, 2009.

[7] P. Cremonesi and R. Turrin. Analysis of cold-start recommendations in iptv systems. In *RecSys*, pages 233–236, 2009.

[8] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38, 1977.

[9] A. Einstein. On the movement of small particles suspended in a stationary liquid demanded by the molecular-kinetic theory of heat. *Annalen der Physik*, 17:549–560, 1905.

[10] Z. Gantner, L. Drumond, C. Freudenthaler, S. Rendle, and L. Schmidt-Thieme. Learning attribute-to-feature mappings for cold-start recommendations. In *ICDM*, pages 176–185, 2010.

[11] Z. Gantner, S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme. MyMediaLite: A free recommender system library. In *RecSys*, pages 305–308, 2011.

[12] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In *SIGIR*, pages 230–237, 1999.

[13] Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. In *ICDM*.

[14] G. Jawaheer, M. Szomszor, and P. Kostkova. Comparison of implicit and explicit feedback from an online music recommendation service. In *HetRec*, pages 47–51, 2010.

[15] D. Kelly and N. J. Belkin. Display time as implicit feedback: understanding task effects. In *SIGIR*, pages 377–384, 2004.

[16] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *KDD*, pages 426–434, 2008.

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

[18] C. Liu, R. W. White, and S. Dumais. Understanding web browsing behaviors through weibull analysis of dwell time. In *SIGIR*, pages 379–386, 2010.

[19] N. N. Liu, E. W. Xiang, M. Zhao, and Q. Yang. Unifying explicit and implicit feedback for collaborative filtering. In *CIKM*, pages 1445–1448, 2010.

[20] Y. Liu, B. Gao, T.-Y. Liu, Y. Zhang, Z. Ma, S. He, and H. Li. Browserank: letting web users vote for page importance. In *SIGIR*, pages 451–458, 2008.

[21] H. Ma, I. King, and M. R. Lyu. Learning to recommend with social trust ensemble. In *SIGIR*, pages 203–210, 2009.

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

[23] P. Melville, R. J. Mooney, and R. Nagarajan. Content-boosted collaborative filtering for improved recommendations. In *AAAI/IAAI*, pages 187–192, 2002.

[24] Y. Moshfeghi, B. Piwowarski, and J. M. Jose. Handling data sparsity in collaborative filtering using emotion and semantic based features. In *SIGIR*, pages 625–634, 2011.

[25] E. R. Núñez-Valdéz, J. M. C. Lovelle, O. S. Martínez, V. García-Díaz, P. O. de Pablos, and C. E. M. Marín. Implicit feedback techniques on recommender systems applied to electronic books. *Computers in Human Behavior*, 28(4), 2012.

[26] M. Papagelis, D. Plexousakis, and T. Kutsuras. Alleviating the sparsity problem of collaborative filtering using trust inferences. In *iTrust*, pages 224–239, 2005.

[27] R. Ratcliff and G. McKoon. The diffusion decision model: Theory and data for two-choice decision tasks. *Neural Computation*, 20(4):873–922, 2008.

[28] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock. Methods and metrics for cold-start recommendations. In *SIGIR*, 2002.

[29] E.-J. Wagenmakers. Methodological and empirical developments for the ratcliff diffusion model of response times and accuracy. *European Journal of Cognitive Psychology*, 21(5):641–671, 2009.

[30] R. W. White and D. Kelly. A study on the effects of personalization and task information on implicit feedback performance. In *CIKM*.

[31] Y. Xin and H. Steck. Multi-value probabilistic matrix factorization for ip-tv recommendations. In *RecSys*, pages 221–228, 2011.

[32] M. Ye, P. Yin, W.-C. Lee, and D. L. Lee. Exploiting geographical influence for collaborative point-of-interest recommendation. In *SIGIR*, pages 325–334, 2011.

[33] T. Zhou, Z. Kuscsik, J.-G. Liu, M. Medo, J. R. Wakeling, and Y.-C. Zhang. Solving the apparent diversity-accuracy dilemma of recommender systems. *PNAS*, 2010.