Immersive Recommendation: News and Event Recommendations Using Personal Digital Traces

Cheng-Kang Hsieh
UCLA CSD
changun@cs.ucla.edu

Longqi Yang
Cornell Tech
ylongqi@cs.cornell.edu

Honghao Wei
Tsinghua University
weihh12@mails.tsinghua.edu.cn

Mor Naaman
Cornell Tech
mor.naaman@cornell.edu

Deborah Estrin
Cornell Tech
destrin@cs.cornell.edu

ABSTRACT

We propose a new user-centric recommendation model, called Immersive Recommendation, that incorporates cross-platform and diverse personal digital traces into recommendations. Our context-aware topic modeling algorithm systematically profiles users' interests based on their traces from different contexts, and our hybrid recommendation algorithm makes high-quality recommendations by fusing users' personal profiles, item profiles, and existing ratings. Specifically, in this work we target *personalized news and local event recommendations* for their utility and societal importance. We evaluated the model with a large-scale offline evaluation leveraging users' public Twitter traces. In addition, we conducted a direct evaluation of the model's recommendations in a 33-participant study using Twitter, Facebook and email traces. In the both cases, the proposed model showed significant improvement over the state-of-the-art algorithms, suggesting the value of using this new user-centric recommendation model to improve recommendation quality, including in cold-start situations.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Information Filtering

Keywords

Personal digital traces, Small data, Personalization, Recommendations

1. INTRODUCTION

With the rise of the web, social media, e-commerce, and mobile communications, individuals generate almost continuous digital traces. From topics referred to in Twitter or email, to web browser history, to digital purchase records, these traces reflect who we are, what we do, and what we are interested in [17]. While the proliferation of these traces accelerates, there has been little exploration of how to utilize digital traces across services to improve the individual user experience. In this paper, we propose a new user-centric recommendation model, called **Immersive Recommendation**, that utilizes individual users' personal digital traces to make recommendations that meet their interests.

As digital devices and services have become extremely pervasive in our daily lives, we propose to capture personal interests from individuals' digital traces, and use this information to make recommendations that satisfy their needs. Our recommendation model is grounded in **interest development theory**, which argues that people develop interests in fields that they are actively involved with; and a person often develops different interests as they play different roles in life [20]. In contrast to the traditional provider-centric model, immersive recommendation is a user-centric model driven by the individual user as the common denominator, and beneficiary, of access to their data, and empowers users to benefit from their data across multiple services in a way that single service provider may not achieve.

To illustrate this idea, in this work we target *personalized news and local event recommendations* initially for their utility and societal importance [18, 23, 36], and leverage users' social media records and personal email communications to make news and local meetup event recommendations. As illustrated in Figure 1, our recommendation process is divided into two phases, user profiling and recommendation. In the user profiling phase, we infer users' interests from their digital traces and create a user profile that has strong predictive power to the kinds of items the user will be interested in. In the recommendation phase, we use a novel hybrid collaborative filtering algorithm to make recommendations by using both user/item profiles and the existing ratings.

Learning interests from users' digital traces poses significant challenges since, in the raw, these traces are replete with context-specific noise that could overwhelm the users' interests within the domain of news or events (or other applications to which this model will be applied in the future). As such, in the profiling phase, a novel unsupervised *Channel-Aware Latent Dirichlet Allocation* (CA-LDA) is proposed to infer the topics a user is frequently engaged with while suppressing the context-specific noise to focus on the users' relevant interests. For example, given an email in which the user scheduled a meeting with colleagues to discuss the upcoming CES exhibition, the CA-LDA will ignore the email-specific topics, such as "meeting" and "scheduling", and focus on the salient topic of the mail, i.e. the CES exhibition.

Through this model, we project users' digital traces into a K-dimensional topical embedding and define a user profile to be the sum of these projections weighted by each instance's potential relevance to the user's interests. In the recommendation phase, we propose a new hybrid collaborative algorithm to fuse the machine-generated user/item profiles with the human-generated rating information to predict a user's preference to each item that she has not

WWW 2016, April 11–15, 2016, Montréal, Québec, Canada.
ACM 978-1-4503-4143-1/16/04.
http://dx.doi.org/10.1145/2872427.2883006 .

rated before. We introduce a latent offset on top of each user/item profile to capture the preference signals that are not captured by the profile but manifested in the ratings. This model ensures that the recommended items are not just relevant to the users' interests, but also have a high chance to satisfy users' needs and expectations, and the recommendations can be fine-tuned to meet a user's specific interests on the target platform.

We evaluated the feasibility and efficacy of the proposed approach with a large-scale offline evaluation. To evaluate the recommendations for news and for local events we used large-scale datasets of Medium.com and Meetup.com users. We generated a profile for each user with their public Twitter traces and compared the discriminative power of the proposed CA-LDA model with that of the standard LDA and *doc2vec* [26]. The result showed that CA-LDA outperformed the previous algorithms by up to 77.4% in terms of mean Average Precision (mAP) in discriminating items the user liked and did not like in this specific task.

We further evaluated the end-to-end performance of the proposed recommendation model by measuring how accurately it can predict a user's preferences in different phases of the user's life time. We compared our approach with a popularity-based baseline, probabilistic matrix factorization (PMF) [32], and collaborative topic modeling (CTM) [44], which is one of the the state-of-the-art algorithms for content recommendations. The results were promising: our approach outperformed the other algorithms by up to 57.9% in both average Recall@50 and mean reciprocal rank (MRR) during both the user-cold-start and the post-user-cold-start phases. The recommendations our algorithm made to a new user were even more accurate than the recommendations other algorithms were able to make after they included up to 10 feedback signals from the users; and the accuracy of our algorithm kept improving with more user feedback.

We also conducted a 33-person within-subject user study to evaluate the utility of the proposed model in an interactive setting with direct user evaluation. For each participant, we used at least one of email, Facebook, and Twitter traces to generate news and meetup recommendations, and compared the utility of recommendations relative to those generated by the above-mentioned algorithms. Our approach showed statistically significant improvement over the other algorithms in 6 out of 8 cases.

Finally, we developed a real-world web application, called *Newsfie*[1], to publicly demonstrate the practicability of the proposed recommendation model, and incorporate a wider range of personal data sources, including watch history on Youtube, and team communications on Slack. The application will support future user studies to understand more qualitative aspects of the recommendation performance and the extent to which these additional data streams further improve performance and user experience.

In summary, the contributions of this work are as follow:

- We propose *Immersive Recommendation*, a new user-centric recommendation model that leverages users' diverse personal digital traces to make recommendations on the user's behalf. To our knowledge, this is the first work to study personal, cross-platform, news and local-event recommendations based on individual-user's multi-channel digital traces.
- We propose a novel profiling algorithm, and a recommendation algorithm to address the unique challenges of Immersive Recommendation.
- We conducted a large-scale offline evaluation, a small user study, and the real-world service deployment to explore the feasibility, efficacy, and practicability of this new recommendation model for two key application domains. The results suggest promising benefits of leveraging users' digital traces to improve future recommender systems and, at the same time, suggests that further research is needed to refine the techniques through real world experiments in order to realize the full potential of immersive recommendations.

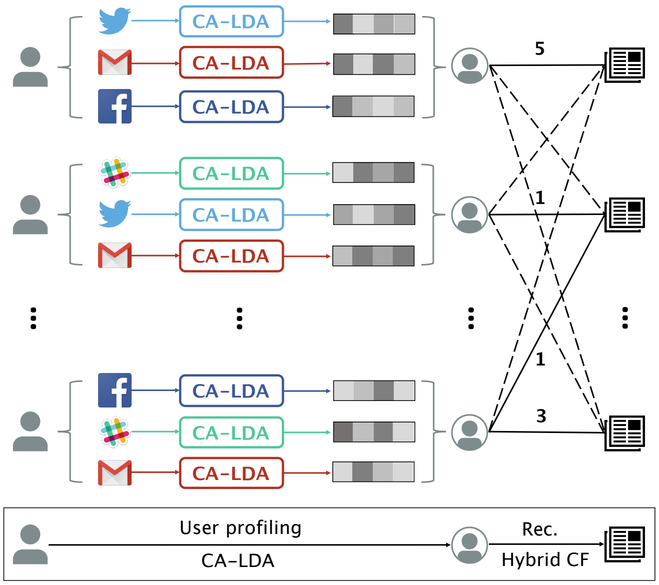


Figure 1: An overview of immersive recommendation. In the profiling phase, Context-Aware-LDA (CA-LDA) is proposed to systematically profile users' digital traces from different contexts and create user profiles. In the recommendation phase, a hybrid collaborative filtering algorithm is proposed to fuse the user profiles, the item profiles, and the existing ratings to predict the ratings that are still unknown.

2. PROBLEM DEFINITION

In the following subsections we formally define the two phases of immersive recommendation, user profiling and recommendation.

2.1 User Profiling

Given a user i, the input of the user profiling problem is a set of digital trace instances denoted as:

$$\mathbf{N}_i = \left\{ n_{i,m} = (c_{i,m}, E_{i,m}), m = 1, ..., M \right\},$$

where $n_{i,m}$ represents each instance in the user i's digital traces. $n_{i,m}$ can be an email thread, a set of relevant tweets or a series of Facebook posts. $c_{i,m} \in \mathbf{C}$ denotes the context in which the instance was generated, such as Email, Twitter, or Facebook, and $E_{i,m}$ is the content of the instance. In this work, we focus on the textual content of the digital traces for its availability in both social and personal context. Therefore $E_{i,m}$ can represent the text in an email thread or in the social media records. The goal of the user profiling is to create a function that transforms $\mathbf{N}_i$ into a feature vector $\mathbf{u}_i$ that characterizes user i's interests. Similarly, each item j for recommendation is also associated with an item profile $\mathbf{v}_j$ derived from its contents.

[1] http://newsfie.org

2.2 Recommendation

Given I users and J items, the input to the recommendation problem are user profile $\mathbf{u}_{i=1...I}$, and item profile $\mathbf{v}_{j=1...J}$, and the existing ratings user i gave to item j denoted as r_{ij}. For each user i, the goal of the recommendation task is to predict the unknown rating user i will give to an item j that she has not rated before. The effectiveness of a recommendation algorithm is evaluated by the top-N recommendation tasks, where the N items with the highest predicted ratings will be recommended to the user, and the user's rating to the recommended items will be used to evaluate the recommendation accuracy.

We are concerned with the recommendation performance in two scenarios. The first scenario, *user-cold-start*, makes recommendations to a new user accessing the system for the very first time. The second scenario, *post-user-cold-start scenario*, makes recommendations for an existing user who has already rated some items.

In the following section we describe the proposed methodology to the user profiling and recommendation problems.

3. PROPOSED METHODOLOGY

We propose an algorithm based on topic modeling to analyze users' digital traces and generate a profile for each user. In the recommendation phase, we use a latent factor based algorithm to predict the rating each user will give to each item.

3.1 User Profiles with Topic Modeling

Our user profiling strategy builds on the premise that a person tends to be interested in the topics she is engaged with in her daily life [20]. We use topic modeling techniques to characterize this engagement from the user's digital traces. Given a collection of documents, topic modeling provides a human-interpretable low-dimensional representation for documents [11]. Topic modeling was originally designed for corpus exploration, and has been extended to other applications, including profiling item contents in a recommender system [11] and categorizing users' social media feeds [10].

However, in contrast to the prior work that focuses on modeling data in a single platform, we deal with data from multiple different contexts and for various platforms. Directly applying single-corpus techniques to this problem may result in poor performance, as shown in our experiment. As such, we propose **Context-Aware LDA** (CA-LDA), a topic modeling algorithm that is able to perform cross-platform modeling to simultaneously model user-data from multiple contexts and item-contents of multiple domains.

Such a topic-model-based approach has three major advantages for immersive recommendation: First, it is fully unsupervised and can be easily extended to analyze user-data from a new context, or new types of items, without costly human-labeling process or hand-tuning. Second, in contrast to other representation learning algorithms, such as *doc2vec* or matrix factorization [26, 52], the user profile based on topic modeling is semantically-meaningful. This is an important feature to allow for recommendations that are more transparent and trustworthy to the user [11, 43]. Third, this approach does not use personal data to train the model. The trained model can then perform inference entirely on the client-side to mitigate privacy concerns of immersive recommendation (as demonstrated in [1]).

Next, we provide a quick introduction to topic modeling, then describe how we apply it for immersive recommendations.

3.2 Latent Dirichlet Allocation (LDA)

A widely-used topic modeling algorithm is Latent Dirichlet Allocation (LDA). Given a corpus of D documents and a vocabulary of size V, LDA assumes there are K topics in the corpus, each of which is characterized by a word distribution $\phi_k \sim \text{Dirichlet}(\beta)$. LDA assumes the following generative process for each document d:

1. Draw a topic distribution $\theta_d \sim \text{Dirichlet}(\alpha_{1...K})$
2. For each word n in document d,
   (a) Draw a topic assignment $z_{d,n} \sim \text{Mult}(\theta_d)$
   (b) Draw a word $w_{d,n} \sim \text{Mult}(\phi_{z_{d,n}})$

With a sparse θ_d (controlled by $\alpha_{1...K}$), words that co-occur in many documents tend to be assigned the same topic. The word distribution of each topic reveals different themes underlying a corpus while the topic distribution θ_d of a document characterizes the themes the document is associated with. From an embedding point of view, θ_d is document d's projection in a low-dimensional non-negative topical embedding [7]. Two documents associated with the similar themes will be projected to points that are proximate to one another in this embedding.

3.3 Basic LDA Profiler

Given a user's digital traces $\mathbf{N}_i$, one straightforward way to use LDA for user profiling, referred to as Basic LDA, is as follows:

1. Train an LDA model with the item corpus (e.g. the news articles or meetup group descriptions)
2. Use the trained model to infer the topic distributions for each instance $n_{i,m} \in \mathbf{N}_i$ denoted as $\theta_{i,m}$
3. Define the user profile $\mathbf{u}_i$ as the weighted sum of the topic distributions of $\theta_{i,m}$ based on each instance's potential relevance to the user's interests. [2]

This approach defines a profile $\mathbf{u}_i$ to be the center of mass of $n_{i,m}$'s projections in the topical embedding created with the item corpus, and an item whose topic distribution θ_d is closer to $\mathbf{u}_i$ is supposed to be associated with the themes that are frequently mentioned in the user's digital traces. While intuitive, this approach has two major problems:

Insufficient coverage: The trained model is not able to cover diverse language usage in the users' digital traces particularly when the item corpus is relatively small. For example, an LDA model trained with meetup descriptions from Meetup.com shows rather poor performance in profiling users' interests (see Section 4.2). This is not only due to a smaller number of items available in the meetup corpus, but also because the meetup descriptions are much shorter and narrower in topic and vocabulary.

Context-specific noise: The other issue is that the user profiles generated by Basic LDA are biased towards the context-specific topics that prevail in a certain context but do not represent users' interests. For example, in email, people tend to use words, "discuss", "meet", etc., that are often classified as office- or work-related themes. It is usually the case that a person mentions these terms not because she is interested in the office-related topics, but because email is usually used in work-related contexts. In other words, the occurrences of these words are largely *independent* of the user's interests and should be excluded from the user's profile. Such context-specific noise exists in many kinds of traces. For example, on Twitter, people tend to have "share", "love",

[2] See Section 4.2 for more on the weighting scheme.

"video", etc. social-oriented terms, but they are rarely associated with users' real interests. When we directly use the topic distributions learned by LDA, this noise overwhelms the real interests of the user.

3.4 Context-Aware LDA Profiler

To address the above-mentioned problems, we propose Context-Aware LDA (CA-LDA). This model originates from the techniques used in the *comparative text mining* [34, 51], where multiple corpora are co-trained in a single model to reveal the commonalities and discrepancies between them. In CA-LDA, we co-train multiple item corpora (news and meetup descriptions) along with the digital trace corpora (Twitter, Facebook, and email). CA-LDA assumes that all the corpora share a superset of *salient topics*, i.e. the topics that reflect users' interests, and each corpus individually has its own unique *background topic* that is associated with the context-dependant noise.

The intuition here is that, given a large number of trace instances from a certain context, the context-dependant noise should prevail in these instances regardless of their main topics. To model this intuition, we assume each document to be a mixture of salient topics and the background topic, and the background words are sampled directly from the word distribution of the context's background topic independent of the document's topic distribution θ_d. Specifically, for each corpus c and the documents in it, the Context-Aware LDA assumes the following generative process:

1. Draw a word distribution $\phi_c \sim$ Dirichlet(β_c) for the background topic
2. For each document d in corpus c,
   (a) Draw salient words proportion $\lambda_d \sim$ Beta($\gamma_\alpha, \gamma_\beta$)
   (b) Draw topic distribution $\theta_d \sim$ Dirichlet($\alpha_{1...K}$)
   (c) For each word n, draw $x_{d,n} \sim$ Bin(λ_d)
      i. If $x_{d,n} = 1$ (a salient word)
         A. Draw a topic assignment $z_{d,n} \sim$ Mult(θ_d)
         B. Draw a word $w_{d,n} \sim$ Mult($\phi_{z_{d,n}}$)
      ii. If $x_{d,n} = 0$ (a background word)
         A. Draw a word $w_{d,n} \sim$ Mult(ϕ_c)

When $x_{d,n} = 1$, the generation of the word is identical to LDA except that the word distributions of salient topics are shared across different corpora. When $x_{d,n} = 0$, the generation of the word is independent from the document's topic distribution θ_d and directly drawn from the corpus-specific background topic. This design makes the terms that prevail in a particular context more likely to be assigned to the background topic and, at the same time, prevents them from diluting the salient topic distribution θ_d. On the other hand, co-training multiple corpora mixes topics in different corpora and allows the smaller corpus (i.e. the meetup descriptions in our case) to benefit from the diverse topics and vocabulary in the larger corpus (i.e. the news articles in our case). The inclusion of a large news article corpus also increases the robustness of the word distribution ϕ_k due to the longer documents and more diverse word choices contained in the news articles.

Similar to the Basic LDA profiler, given a user's digital traces N_i, the trained Context-Aware model is used to infer the topic distribution for each instance $n_{i,m}$. The user profile $\mathbf{u}_i$ is again defined as the weighted sum over the instances' topic distributions $\theta_{i,m}$ [3], but now the $\theta_{i,m}$ are only associated with the salient topics and separated from the context-specific terms. As shown in our experiments, this leads to a user profile $\mathbf{u}_i$ that is much more focused on the user's real interests and has a stronger predictive power.

Context	Background Terms
Email	*pleas, offic, schedul, convers, fax, cellular*
Twitter	*awesom, share, tweet, post, video, love*
Facebook	*love, night, happi, tomorrow, final, tonight*
Medium.com	*happen, idea, actual, experi, hard, reason*
Meetup.com	*social, event, network, singl, profession, join*

Table 1: Frequent background terms learned by Context-Aware LDA. The words were stemmed before included in the model.

We sampled digital traces from the datasets that exist in the public domain to train the model. For example, the Enron email dataset [25] and several public mailing lists were used to construct the email corpus that covers diverse topics; 100,000 users' public Twitter data and 1,200 users' public Facebook posts were used to construct the Twitter and Facebook corpora. The model is implemented based on Mallet's parallel LDA implementation; the source code and the trained model are released at [2]. Some of the learned background terms are listed in Table 1.

In addition to the user profile, we also use CA-LDA to generate item profile $\mathbf{v}_j$ from the item contents, which will be used in the recommendation as well. Compared to other ad-hoc noise suppressing approaches, such as hand-crafting a list of background terms, or carefully tuning the *tf-idf* thresholds for different corpora, CA-LDA resolves the context noise issue in a systematic way. For immersive recommendation, where we need to deal with data from a large and increasing number of different contexts, the CA-LDA's ability to avoid costly hand-tuning is particularly valuable [12].

3.5 Recommendation Phase

Given user profile $\mathbf{u}_i$ and item profile $\mathbf{v}_j$, we are able to identify items that are *relevant* to users' interests. However, the relevance alone is not sufficient in a practical recommendation system. For example, within a large set of items, there may still have a large number of items that are relevant to a user's interests. Further filtering is needed to find the items that will have the highest user-perceived quality. Another issue specific to Immersive Recommendation is that a person's interests would vary from one platform to another; fine-tuning is needed to better match the recommendations to the user's specific interests on the target platform.

We propose a hybrid collaborative filtering algorithm, called **collaborative user-item regression**, that carefully fuses the objective user/item profiles and the subjective rating information to predict the ratings $\hat{r}_{ij}$ that are still unknown. This model is built on the foundation of *regression-based latent factor* [4, 11]. On top of the user profile and item profile, we introduce *latent user offset* $\eta_i \in \mathbb{R}^K$ and *latent item offset* $\varepsilon_j \in \mathbb{R}^K$ to capture the preference information that is not captured by the user/item profile, but manifested in the ratings. The model assumes a generative process for ratings r_{ij} as follows:

1. For each user i, draw user offset $\eta_i \sim \mathcal{N}(0, \lambda_u^{-1} I_K)$
2. For each item j, draw item offset $\varepsilon_j \sim \mathcal{N}(0, \lambda_v^{-1} I_K)$
3. For each user-item pair (i, j), draw the rating

$$r_{ij} \sim \mathcal{N}((\mathbf{u}_i + \eta_i)^T (\mathbf{v}_j + \varepsilon_i), c_{ij}^{-1}). \qquad (1)$$

[3] The proper weighting scheme is learned from the existing users' data. See Section 4.2 for details.

The key ingredient of this model lies in Eq. 1. As $\mathbf{v}_j$ and $\mathbf{u}_i$ are fixed, the free parameters ε_i and η_j will be tuned in a way that the resultant inner product approximates to the rating r_{ij} and makes up the difference between the user rating r_{ij} and the user/item profile relevance, i.e. $\mathbf{u}_i^T \mathbf{v}_j$. This design allows ε_i and η_j to capture the preference information that is missing from the user/item profiles.

Specifically, the item offset ε_j makes up for the hidden characteristics of item j that are not captured by the item profile $\mathbf{v}_j$. Consider article *The Difference Between Living in New York and San Francisco* from Medium.com as an example [14]. Based on the content, this article is not related to technology and thus its profile $\mathbf{v}_j$ has a small weight on the tech-related topics. However, many tech people actually enjoy reading this article and give it high ratings, probably because New York and San Francisco are the two cities where many tech people live or may consider moving to. As these tech people's profile $\mathbf{u}_i$ tend to have a large weight at the tech-related topics, in order to make the inner product in Eq. 1 approximate to the high ratings r_{ij} given by these users, the item offset ε_j will be tuned to have a larger weight in the dimensions that correspond to the tech-related topics, and this, in turn, will make this article more likely to be recommended to the users who are interested in tech (i.e. whose $\mathbf{u}_i$ has a larger weight in the tech-related topics) in the future.

On the other hand, the user offset η_i makes up for the difference between a user's initial profile $\mathbf{u}_i$ and her specific interests on the target platform. To see this, consider a user who gave high ratings to many rock music articles while her initial profile $\mathbf{u}_i$ shows little interest in rock music. Assuming the remaining parameters are fixed, in order to satisfy Eq. 1, the user offset η_i will be tuned to have a larger weight on the rock music related dimension to match the high ratings, and the other articles that are related to rock music will become more likely to be recommended to this user in the future. This adjustment is important for immersive recommendation as a user may have specific interests on the target platform that are not revealed in her profile. For a new user, her offset η_i is a zero vector, and the recommendations are made solely based on her profile. However, once she starts to give ratings, η_i will be tuned to compensate for the difference.

The scale of ε_j and η_i is controlled by the regularization parameters λ_v and λ_u. For example, when λ_u is smaller, a larger η_i is allowed, and the recommendation will lean more towards the preferences expressed in user i's ratings than towards her initial profile $\mathbf{u}_i$.

The precision parameter c_{ij} serves as a confidence parameter for rating r_{ij}, and is set larger if we trust the rating r_{ij} more. This confidence parameter is useful when dealing with implicit ratings [22]. For example, in the case of Medium.com, users only "upvote" the articles they like, and do not have a means to express their dislike for an article. As such, the case that a user did not upvote an article could be interpreted as either **a)** the user did not like the story or **b)** the user was not aware of the story. Therefore, for stories that did not get upvoted, we set $r_{ij} = 0$ and have a lower c_{ij} to capture this uncertainty [22,44], specifically:

$$c_{ij} = \begin{cases} a, & \text{if } r_{ij} = 1 \\ b, & \text{if } r_{ij} = 0 \end{cases} \tag{2}$$

where $r_{ij} = 1$ if user i upvoted item j or otherwise; a and b are tuning parameter, and $a > b > 0$. On the other hand, for ratings that are made on a 1-5 Likert scale, r_{ij} can be set to numbers between 0 and 1 that represent different degrees of support to an item, and c_{ij} can be set accordingly to represent the rating confidence.

3.6 Parameter learning

Since directly computing the posterior distribution of ε_j and η_i is intractable, following Wang and Blei's formulation [44], we use an EM-algorithm [15] to estimate their Maximum A Posteriori probability (MAP). The complete log likelihood of η_i , and ε_j is:

$$\begin{aligned}\mathcal{L} = &-\frac{\lambda_u}{2}\sum_i \eta_i^T \eta_i - \frac{\lambda_v}{2}\sum_j \varepsilon_j^T \varepsilon_j \\ &-\sum_{i,j}\frac{c_{ij}}{2}(r_{ij} - (\mathbf{u}_i + \eta_i)^T(\mathbf{v}_j + \varepsilon_j))^2\end{aligned} \tag{3}$$

We optimize the log-likelihood $\mathcal{L}$ by a coordinated ascent. Let $\hat{\mathbf{u}}_i = \mathbf{u}_i + \eta_i$, $\hat{\mathbf{v}}_j = \mathbf{v}_j + \varepsilon_j$, and $\hat{U} = \hat{\mathbf{u}}_{i=1}^I$, $\hat{V} = \hat{\mathbf{v}}_{j=1}^J$. We take the gradient of $\mathcal{L}$ with respect to $\hat{\mathbf{u}}_i$ and $\hat{\mathbf{v}}_j$, respectively. Setting the gradient to zero gets:

$$\hat{\mathbf{u}}_i \leftarrow (\hat{V} C_i \hat{V}^T + \lambda_u I_K)^{-1} (\hat{V} C_i R_i + \lambda_u \mathbf{u}_i) \tag{4}$$

$$\hat{\mathbf{v}}_j \leftarrow (\hat{U} C_j \hat{U}^T + \lambda_v I_K)^{-1} (\hat{U} C_j R_j + \lambda_v \mathbf{v}_j) \tag{5}$$

where C_i is a diagonal matrix with c_{ij} for $j = 1...,J$ as its diagonal elements and $R_i = (r_{ij})_{j=1}^J$ for user i. C_j and R_j are defined in a similar way for item j.

In each iteration, we update all $\hat{\mathbf{u}}_i$ with the latest estimation of $\hat{\mathbf{v}}_j$, and update all $\hat{\mathbf{v}}_j$ with the latest estimation of $\hat{\mathbf{u}}_i$. This estimation process stops when the log-likelihood $\mathcal{L}$ converges, and the offsets η_i and ε_j can be computed accordingly. The complexity of each iteration is linear to the number of known ratings r_{ij}, but each iteration enjoys a high degree of parallelism as all the $\hat{\mathbf{u}}_i$ and all the $\hat{\mathbf{v}}_j$ can be estimated concurrently.

3.7 Prediction and Updating

After offsets η_i and ε_j are learned, we use our model to predict an unseen rating $\hat{r}_{ij}$ as follow:

$$\hat{r}_{ij} \approx (\mathbf{u}_i + \eta_i)^T (\mathbf{v}_j + \varepsilon_j), \tag{6}$$

where $\hat{r}_{ij}$ is the expectation of the rating given every known rating r_{ij} and every user/item profile.

When a user starts to make new ratings, the new ratings are incorporated into the model and update the user offset η_i by optimizing the $\hat{\mathbf{u}}_i$ with the updated C_i and R_i as in Eq. 4. However, computing $\hat{V} C_i \hat{V}^T$ has time complexity $O(J)$ and is too slow for real-time personalization when the number of item J is large. This, however, can be optimized based on the observation that

$$\hat{V} C_i \hat{V}^T = b \hat{V} \hat{V}^T + (a - b) \sum_{j \in S(i)} \hat{\mathbf{v}}_j \hat{\mathbf{v}}_j{}^T, \tag{7}$$

where $S(i)$ is the set of items user i has upvoted. The optimization is done by caching $b\hat{V}\hat{V}^T$, and only computing $(a-b) \sum_{j \in S(i)} \hat{\mathbf{v}}_j \hat{\mathbf{v}}_j^T$ in the update process. This optimization introduces $\frac{J - |S(i)|}{|S(i)|}$ times of speed-up, which is quite significant as $J \gg |S(i)|$ in most cases.

4. OFFLINE EVALUATION

We conducted a large-scale offline evaluation to study the performance of the proposed profiling algorithm and the collaborative filtering model. We created profiles for individual users based on their public Twitter traces, and used that to predict what news and events they liked on Medium.com and Meetup.com. Note that Twitter data was the only digital trace used in the evaluation as it is one of a few publicly-available yet personally-identifiable data sources that allows for such a large-scale offline study. (This constraint is relaxed in the user study described in Section 5 for both the profile generation phase and the recommendation phase.) In the following, we describe our data collection and profile generation strategies, evaluation methods and the results.

4.1 Data Collection Strategy

Many Medium.com and Meetup.com users declare their Twitter handle on the profile page. We randomly chose 63,053 Medium.com and 50,000 Meetup.com users who declared their Twitter handle and crawled their public Twitter traces to create the profiles [4]. For each of these users, we crawled: (a) their most recent 3,000 public tweets through Twitter API, (b) the tweets made by the people they followed, and (c) the tweets (made by other people) that were associated with the user's hashtags (to capture topics the user paid attention to).

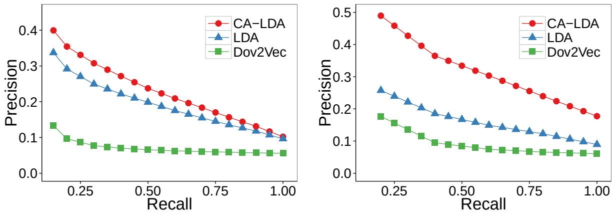
The users' records on Medium.com and Meetup.com are taken as the ground truth for their news and event preferences. For Medium.com, the ground truth is the news articles each user has upvoted. For Meetup.com, the ground truth is the meetup groups in the New York City each user has joined. We crawled 31,000 news stories, and 11,823 meetup groups. On average each Medium.com user has upvoted 13.1 news stories, and each Meetup.com user has joined 5.1 meetup groups. The upvotes and group memberships both follow a long-tail distribution — the top 10% of the most popular items account for 59% and 61% of upvotes and group memberships respectively. These long-tail phenomena had important implications for the recommendation performance as discussed further in the Section 4.3.

4.2 Profiling Performance

We generated a profile for each user based on their Twitter traces. As suggested in [10], rather than treating each individual tweet as an instance, we treated a set of tweets associated with the same source as one instance in order to infer a more robust topic distribution. Specifically, a user's digital trace set $\mathbf{N}_i$ consisted of a unique instance that was composed of the most recent tweets made by this user and multiple instances that were composed of the tweets made by each of her followees, and the tweets associated with each hashtag she used. The followees' tweets allowed us to enhance the profile precision in particular for passive users who posted only few or no tweets [35]. The tweets associated with the hashtags allowed us to better understand the topics the user referred to. For robustness, we randomly selected 300 followees, and 300 hashtags to include into a user's profile.

We defined the user profile $\mathbf{u}_i$ as the weighted sum over the topic distributions of these instances as mentioned in Section 3.4. We determined the proper weighting through a grid search. The first instance, composed of the user's own tweets, was weighted by the number of the tweets in it; the instances made by the followees and those associated with the hashtags were weighted by 5 and 0.2 respectively. The background topic for the Twitter context (i.e. $\phi_{twitter}$) was used to filter the Twitter-specific background noise in all the instances. We only included words that have more than 3 characters and stemmed the words before including them in the model. The stopwords and URLs were excluded.

The item profiles $\mathbf{v}_j$ were generated in a similar fashion. The topic distributions of the article or the meetup description computed with the corresponding background topic (i.e. ϕ_{medium} or ϕ_{meetup}) was taken as the profile for each item.



(a) Medium.com

(b) Meetup.com

Figure 2: Precision-recall curves for different profiling algorithms. CA-LDA outperformed the prior algorithms by 18.7% and 77.4% in mean average precision for Medium.com and Meetup.com respectively.

4.2.1 Evaluation Strategy

We randomly chose 3,000 Medium.com and Meetup.com users to test the profiling performance. As in [35], for each user we created a set of items $\mathbf{J}$ composed of items the user liked (denoted as $\mathbf{J}_{like}$) and the user did not like (denoted as $\mathbf{J}_{dislike}$). A good profiling algorithm should be able to discriminate $\mathbf{J}_{like}$ from $\mathbf{J}_{dislike}$ based on the similarity between the user profile and the item profile generated by the algorithm. Considering the average number of upvotes and group memberships per-user, for Medium.com users, we randomly chose 10 liked articles as positive examples, and 190 articles they did not like as negative examples. For Meetup.com users, we chose 5 meetups each of them joined, and 95 meetups they did not join. We ranked the items in $\mathbf{J}$ by the profile's *cosine similarity* to the user's profile and computed the prediction precision for different recall rates.

We compared the profiling performance of the proposed CA-LDA algorithm with that of the standard LDA, and *doc2vec*, which is a state-of-the-art text representation learning algorithm based on neural networks [26]. We used the LDA implementation from Mallet [31] with $\beta = 0.01$ and $\alpha_i = \frac{1}{K}$ for i=1,2,...,K. CA-LDA has additional parameters: $\beta_c = 0.1$ and $(\gamma_\alpha, \gamma_\beta) = (0.2, 0.8)$. The doc2vec implementation from *Gensim* was used with the default parametrization [37]. We tested different model sizes for $K = 50, 100...500$, and only presented the results for $K = 200$ where the performance of all three algorithms saturated. The same weighting scheme is used across different algorithms.

4.2.2 Evaluation Results

Figure 2 shows the average precision and recall curves with different profiling algorithms. CA-LDA outperformed both LDA and doc2vec in every case. Specifically, for Medium.com, CA-LDA had 18.7% improvement in mean Average Precision (mAP) over LDA due to its ability to focus on the salient topics referred to in users' Twitter traces and in the item contents. For Meetup.com, CA-LDA had a much more significant (77.4%) improvement over LDA that was trained with only the meetup description corpus. In addition to the above-mentioned benefit, this demonstrated the advantage of co-training multiple corpora to allow a smaller corpus (i.e. the meetup description corpus) to benefit from the richer linguistic features contained in the others. In general doc2vec had a much poorer performance compared to LDA-based algorithms due to its sensitivity to the location of the words in a document (unlike LDA's bag-of-word model) that made the model trained with text in one context less-generalizable to the text in another context in this specific task [26].

The result above showed that CA-LDA can effectively learn users' interests from their Twitter data. We further evaluated the

[4] These users were chosen from 300,000 Medium users discovered through Medium.com's upvote graph and about one million Meetup.com users based in the New York City.

mAP	Own	Followees	Hashtags	Combined
Medium	0.237	0.240	0.188	0.245
Meetup	0.331	0.347	0.282	0.353

Table 2: The predictive power of different types of tweets using CA-LDA.

predictive power of different types of tweets. Table 2 reports the mAP of CA-LDA when different types of tweets were used individually, and when they were combined. Interestingly, the followees' tweets were the most informative among all three types while the hashtag tweets were the least, which was consistent with the results in [35]. The combination of the tweets only showed marginal improvement. This could be because the inherent correlation among these three signals [35]. In the future work, we will explore if more sophisticated weighting schemes can further improve the performance.

In addition, we found that the length of Medium.com articles was also indicative of the users' preferences and could additionally improve the CA-LDA's mAP by 13%, but the length of the meetup descriptions did not have such an effect. This was probably because the length of a news article is more related to the article's quality. We also evaluated the performance of other kinds of information retrieval models. For example *BM25*, one of the classical text-retrieval algorithms based on probabilistic relevance framework [38], showed performance comparable to CA-LDA for Medium.com and had about 7% lower mAP than CA-LDA for Meetup.com. However, it is unclear how to combine the BM25 scores with the user-rating information, which is crucial to the performance of a practical recommendation system as we will see in the next subsection.

4.3 Recommendation Performance

After validating the predictive power of the user profile, we conducted a large-scale recommendation task to measure the end-to-end performance of the immersive recommendation model (**ImmRec**). We evaluated the recommendation performance for Medium.com users and Meetup.com users, and compared the recommendation accuracy with the following prior approaches:

1. Content-Based (**CONTENT**) ranks the items by cosine-similarity between the user/item profiles that CA-LDA learned. This algorithm represents pure content-based models. Whenever a user likes an item, the profile is updated by adding the item profile $\mathbf{v}_j \times \alpha$, where the smoothing parameter α is set 0.1 for the best performance.

2. Most Popular First (**POPULAR**) ranks the items by number of upvotes or members an item had, which is a common baseline for the user-cold-start problem. While simple, this algorithm is a strong baseline for comparison in the case of news and events recommendations as the majority of upvotes or group memberships are concentrated in a small subset of items.

3. Probabilistic Matrix Factoring (**PMF**) makes recommendations without user or item profiles [32]. PMF represents traditional user-user collaborative filtering models.

4. Collaborative Topic Modeling (**CTM**) uses PMF along with item profiles to improve the recommendations [44]. CTM is one of the state-of-the-art document recommendation models, and represents existing hybrid recommendation algorithms in use.

We set $K = 200$, $a = 1$, and $b = 0.01$ for all the collaborative filtering based algorithms, including ImmRec, PMF, and CTM, and set $\lambda_u = \lambda_v = 10$ for ImmRec, $\lambda_u = \lambda_v = 0.01$ for PMF, and $\lambda_u = 0.01, \lambda_v = 10$ for CTM according to the prior work [11,32].

4.3.1 Training/Testing Data Segregation

We randomly chose 5,000 Medium.com and 5,000 Meetup.com users to create testing user sets whose upvotes or membership information was excluded from the model. The models for Medium.com and Meetup.com were trained with the rest of the users' data. We evaluated the recommendation accuracy for the testing users starting with the time at which they had made no feedback (i.e. user-cold-start phase) up to the time that they had made 10 upvotes or had joined 5 groups. For Medium.com, we included the upvotes each testing user made **before** January 1st 2015 in a chronological order, and made recommendations only for the stories published **after** January 1st 2015 and before October 1st 2015, which were 22,863 stories in total. This segregation was to ensure that the recommendations were made based on the same pool of items in every case. For meetups, where such temporal information was not available, we randomly chose 2,000 meetups for recommendation and included users' ratings to the rest meetups in a randomized order.

4.3.2 Evaluation Metrics

We generated the top-50 news stories and meetup groups based on the predicted ratings $\hat{r}_{ij}$, and computed the following metrics based on the items the users actually upvoted or joined [45].

- **Average Recall Rate**: Recall rates measure the proportion of positive items that the algorithm was able to identify in a top-M recommendation task. The recall rate for each user is defined as below:

$$\text{Recall}@M = \frac{\text{number of items the user liked in top M}}{\text{total number of items the user liked}}.$$

- **Mean Reciprocal Rank** (MRR): MRR measures the ranking of the first correct item and averages over all the users. This measure provides insight into the ability of the algorithm to return a correct recommendation at the top of the ranking. It is defined as follows:

$$\text{MRR} = \frac{1}{|I|}\sum_{i=1}^{I}\frac{1}{\text{rank}_i},$$

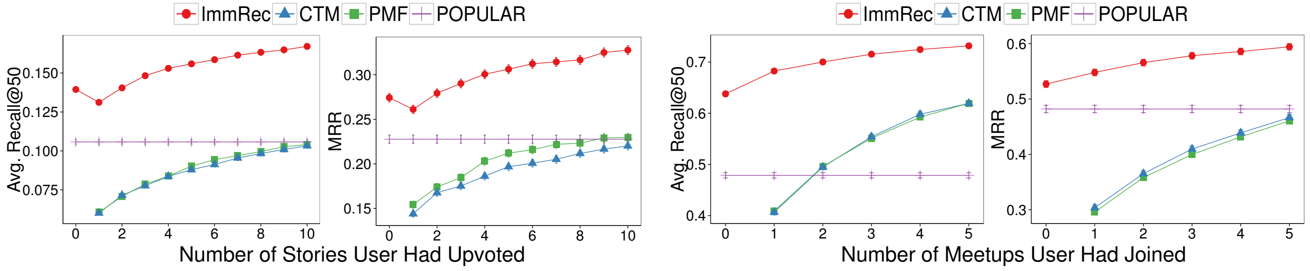
where rank_i is the rank of the first correct item for user i.

Note that, as in most prior work, we were not able to compute the recommendation precision. This was because when a user did not upvote or join an item, we do not know if it was because she did not like it or because she was not aware of it [44]. This drawback was addressed in the user study described in Section 5.

4.3.3 Evaluation Results

The evaluation results in terms of the Recall and MRR are shown in Figure 3. Note that the content-based algorithm (CONTENT) performed over 5x and 3x worse than the popularity-based baseline (POPULAR) for Medium.com and Meetup.com respectively in both cold-start and post-cold-start phases. It is not shown in the figure and omitted hereafter to focus the discussion, but its poor performance underscored the importance of incorporating rating information into the recommendations.

Compared to the remaining algorithms, ImmRec significantly outperformed every other approach in both average recall rate and MRR for Medium.com users, and maintained at least a 14.7% to



(a) Medium.com

(b) Meetup.com

Figure 3: Average Recall@50 and Mean Reciprocal Rank when a user had made 0 to 5 or 10 feedback signals. Immersive recommendation (ImmRec) significantly outperformed the second best approach in every case by up to 57.9% in Recall@50 and 42.6% in MRR, and was able to smoothly improve the performance when more feedback was available (post cold start).

42.6% margin over the second best algorithm. For example, when a user had not made any upvote (i.e. in the user-cold-start phase), our approach was able to make recommendations that were even more accurate than the recommendations PMF and CTM were able to make after 10 upvotes (post cold-start). According to the data we collected, it would take an average user 261 days to make that many upvotes. The results also demonstrated the ImmRec's fine-tuning performance. When a user started to make upvotes, the algorithm was able to incorporate these signals and smoothly improve the recommendations over time. One exception was when a user made fewer than 2 upvotes. In those cases, the user offset η_i leaned too much towards the profile of those few items, and degraded the overall accuracy. This drawback can be addressed by putting a smoothing coefficient in front of η_i in Eq. 6 when the number of upvotes is small, or more systematically, by learning the η_i at different stages from the data as in [50].

Another noteworthy result is the superior MRR of ImmRec. ImmRec's MRR was always the highest among all the algorithms while other collaborative filtering algorithms (i.e. PMF, CTM) were not able to surpass the baseline until later in the user's lifetime. This demonstrates ImmRec's ability to push the relevant items further up into the ranking, which is an important requirement for many recommendation systems [41].

The results for meetup recommendation followed a similar trend. As shown in Figure 3b, the ImmRec outperformed the second best algorithm in every case by a 9.3% and to 42.5% margin. However, the gap between ImmRec and PMF or CTM shrunk quickly particularly in the recall rates. This was probably due to the fact that a user usually had much narrower preferences in terms of joining meetups than reading news articles (manifested in an 5.1% lower average topical entropy according to our topic model). Therefore, PMF and CTM could quickly learn a user's meetup preferences with only a small amount of feedback. Even so, the high-quality cold-start recommendations and a high MRR still make ImmRec a more desirable choice.

5. USER STUDY

We conducted an initial, small user study (N=33) to explore the utility of immersive recommendation in an interactive setting [40]. Moreover, in this study we included not only users' Twitter data, but also Facebook and email traces. The personal communication in email is worthy of exploration for its potential representation of a broader range of interests than what is expressed in social media alone, where impression and audience management dominate [21, 30].

The goal of the experiment was to compare the performance of ImmRec to other algorithms using direct evaluation by users of the recommendations given by each. As in the offline evaluation, we studied the ImmRec's performance for both the user-cold-start and post-user-cold-start phases and compared its performance to that of Most-Popular, Random, PMF, and CTM. To avoid bias due to between-subject differences [40], we adopted a within-subjects design, in which each participant rated items recommended by each of the algorithms, and the performance of the algorithms were compared on a per-user basis [24, 40].

5.1 Experiment Design

The experiment consisted of two sessions, one for news and one for meetup. Each session consisted of two phases: a Cold-Start phase and a Steady-State phase. In the **Cold-Start Phase**, we compared three algorithms: ImmRec, Most-Popular, Random. We presented to the user the top six recommendations generated by each algorithm[5]. The users were not told which items were recommended by which algorithm. As a treatment to the carryover effect, where items presented earlier cause bias in rating later items, the recommended items were presented one at a time, in a randomized order. If two algorithms' recommendation coincide, the same item would only be presented once. For each item, the participants were asked to specify how interesting the presented item is on a 1 to 5 Likert scale [27]. The descriptions of the levels were assigned based on [16] from "Not at all interesting" to "Extremely interesting".

In the **Steady-State Phase**, we used some of the ratings provided by the participant in the first phase, and compared four algorithms: ImmRec, Most-Popular, PMF, and CTM. We presented the top six recommendations from each to each participant in the same fashion as described above. In this phase, of course, the algorithms (other than Most-Popular) could generate recommendations based on the ratings the participants made earlier. Since recommendations made by ImmRec earlier would contain information about the user's profile, for these previous ratings, we only used the ratings for the items recommended by the Most-Popular and Random algorithms in order to prevent other algorithms from benefiting from this information. For a Likert scale rating $l = 1$ to 5, we set $r_{ij} = 0.25 \times (l-1)$, and $c_{ij} = 1$ to update the recommendation models as described in Section 3.7.

The participants were asked to use a web app to connect to at least one of their Gmail, Facebook, and Twitter accounts for the system to access their traces. For email, we removed the signa-

[5] We observed significant fatigue in a pilot study when more than six recommendations from each algorithm were shown.

tures and treated each email thread the user sent or forwarded as an equally-weighted instance. For Facebook, similar to the strategy for Twitter (Section 4.2), we treated all the posts made by the user as one instance weighted by the number of posts in it, and those made by each Facebook Page the user liked as one instance weighted by 5 due to their similar natural as Twitter followees. The profiles from the three different contexts were normalized to their $L1$-norm and summed together. Due to limited data access, we were not able to further fine-tune the parameterization prior to the experiment (in particular for email); this is an important area for future research. For the recommendations themselves, we used 13,250 articles from Medium.com published between July 1st 2015 and October 1st 2015 (news), and all 11,823 meetup groups in the New York City (meetup).

The participants were recruited through mailing lists and flyers on a university campus in New York City. The participants included 1 faculty member, 5 staff members, 2 undergraduates, and 25 graduate students. The study was approved by Cornell Institutional Review Board Protocol #1507005739.

5.2 Evaluation Metrics

We performed a robust examination of the experiment results using numerous strategies for treating the rating data and for the statistical analysis of the differences between conditions. For lack of space, we focus on one set of strategies here, and note the alternative methods when relevant.

As a common treatment to the ordinal data, we first transformed each level in the Likert scale to a value between 0 and 1 using their average cumulative proportion [6], given by:

$$a_j = \sum_{k=1}^{j-1} p_k + 0.5 p_j,$$

where p_j is the proportion of level j among all the ratings. This step is critical as the differences between two consecutive levels were not necessarily uniform. For example, in prior studies of movie recommendation, users had a higher chance to switch their ratings between 2 and 3 than to switch their ratings between 4 and 5, which indicated that there was a larger user-perceived distance between 4 and 5 [8]. We devised the transform schemes for the news articles and meetups separately based on all the users' ratings. The ratings of news stories showed the similar phenomenon as in movie recommendations described in [8], while the meetup ratings were more uniformly distributed. For robustness, we performed the analysis using other functions, including directly using the raw 1–5 Likert scale, or cutting off the ratings at 3.5 and assigning binary scores of 0 and 1 [19]. We found similar results with these alternative functions.

For each participant p, we took $U_{p,a}$, the average of the (transformed) ratings that the participant gave to algorithm a's recommendations, as the *utility* of that algorithm for the participant. We used $U_{p,a}$ to compare the different algorithms in the analysis below.

In addition, to assess the scale of the average improvement of ImmRec, for each competing algorithm a, we computed the average utility improvement ImmRec had over algorithm a across all the users, and normalized it by the respectively global mean of news and meetup recommendations based on all the users' ratings, which is:

$$\frac{1}{M|P|} \sum_{p} (U_{p,ImmRec} - U_{p,Other}),$$

where global mean $M = 0.48$ and 0.47 for news and meetup respectively. We used the improvement metric in the figure below.

5.3 Evaluation Results

We compared the recommendation performance using the metrics described above. The results, in terms of the improvement, are shown in Figure 4. The figure shows the performance improvement of ImmRec over the other algorithms in different settings (news on top, in green; meetup on bottom, in orange) and in cold-start (left) and post-cold-start settings. For example, the left-most bottom bar shows that ImmRec improved over Most-popular by 20.7% in the cold-start Meetup settings. Note that the Random algorithm had an 2x worse performance than the ImmRec and so is not shown in the figure and omitted hereafter to focus the discussion.

We used the $U_{p,a}$ scores to evaluate whether ImmRec outperformed the other algorithms. The statistical significance of the improvement was evaluated using the paired Student's t-test[6] as suggested in [40], and the effect size was evaluated using Cohen's d [13]. [7]

As demonstrated in Figure 4, ImmRec had an improvement in utility over the other algorithms in every case. In the cold-start phase for the meetup recommendation, ImmRec had the greatest improvement. The improvement was statistically significant with $p = 0.00018$ over the Most-Popular algorithm, and the effect size Cohen's d was 0.78.[8] This large improvement was probably due to the fact that people tend to join meetups that are closely related to their daily activities or major interests, which can be more effectively learned from their digital traces (in relative to the more diverse interests in news articles). On the other hand, a smaller improvement (6.3%) was observed in the cold-start phase for the news recommendation. This low performance was probably due to the fact that extremely popular news articles, while not directly relevant to a user's interests, may still have a high chance to appeal to the user. Even so, ImmRec was still able to consistently outperform the Most-Popular algorithm with statistical significance $p = 0.009$ and effect size $d = 0.50$.

Across the board, ImmRec performed better than all other algorithms, though in two cases (compared to CTM and PMF in the steady-state phase for Meetup recommendation) the performance improvement was not statistically significant. This result is consistent with the results we observed in the offline evaluation, where the advantage of ImmRec shrunk quickly after the users made more feedback (see Section 4.3.3). It is also noteworthy that, in our analysis, only ImmRec showed statistically significant improvement over the Most-Popular baseline, while CTM and PMF did not for either news or meetup recommendations. Future work can use human ratings to verify that the performance characteristics would remain for items that are beyond the top six results explored in this study, as suggested by the analysis in Section 4.3.3.

5.4 Error Analysis

Through a closer examination of the cases where ImmRec performed relatively poor, we observed a strong correlation between the topical entropy of a user's profile[9] and the utility of ImmRec in the cold-start phase (r=0.49, $p = 0.003$) while no correlation was

[6] We also used ANOVA for repeated measures and Wilcoxon signed-rank test and found similar results.

[7] Cohen's $d = \frac{\bar{X}_d}{S_d}$, where $\bar{X}_d$ and S_d are the mean and standard deviation of the differences between two algorithms' performances.

[8] An effect size $d = 0.5$ is considered to be a medium effect, and $d = 0.8$ is a large effect.

[9] Topical entropy is defined as $H(\mathbf{u_i}) = -\sum_k p(u_{ik}) \log p(u_{ik})$

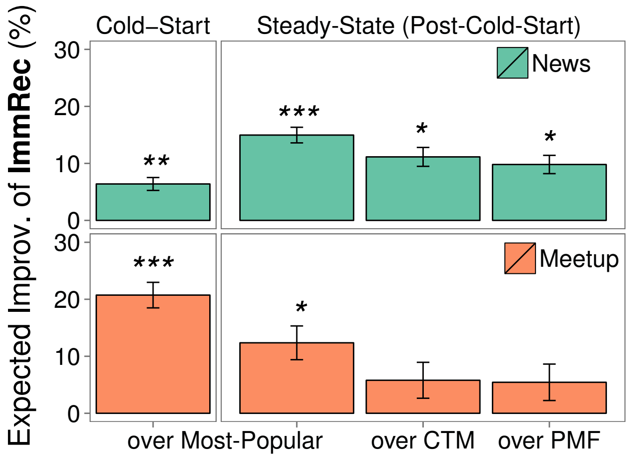


Figure 4: Expected improvement of ImmRec in utility over other algorithms in the user experiment (*N*=33). ImmRec outperformed other algorithms by up to 20.7%, but had less significant improvement during the cold-start phase for news and the steady-state phase for meetups. (Asterisks represent the significance level of the paired t-test. *: *p*<0.05, **: *p*<0.01, ***:*p*<0.001)

observed in the steady-state phase. The topical entropy of the profile also strongly correlated with the number of different types of traces used to generate the profile ($r = 0.70$, $p < 0.001$). These correlations suggested that when a user's profile was biased towards few topics (i.e low entropy), ImmRec's recommendations tended to be less satisfactory for some users in the cold-start-phase, and ImmRec was able to mitigate this issue in the steady-state phase through the fine-tuning. We suggest that practitioners elicit as many diverse traces as possible from users and mix cold-start recommendations with some popular items in order for ImmRec to adjust for the potential discrepancies between the user profile and the user's interests. In future work, we will explore whether more sophisticated weighting schemes can increase the profile entropy. Moreover, questions, such as what kinds of traces contribute most to the recommendations, and how the amount of data changes the recommendation performance, will be explored in the future study as well.

6. RELATED WORK

Broadly, immersive recommendation falls into the category of the recommender systems that use side information, beyond the user-item matrix, to improve the recommendation quality [42]. Prior work has considered information including item contents [11], social network graph [29, 48], user contributed data, such as comments [9] or images [28, 47, 53], and user attributes, such as gender and age [4, 33] (see [42] for a comprehensive survey). However, little prior work considered the recommendation problem from the angle of the individual users, who have access to almost continuous digital traces, spanning professional and personal communication and activities. Immersive recommendation approaches the recommendation problems from this user-centric perspective and empowers individuals to use their own diverse data to improve the quality of the recommendations they receive.

Immersive recommendation is a generalization of the prior work on cross-platform recommendation, where user data in one platform is used to improve recommendations on another platform. For example, prior work used social media records to recommend Pinterest boards [49], Youtube videos [46], and ebooks [39], or aggregated the user profile across different platforms to streamline the on-boarding process on a new platform [3]. However, whereas most prior studies focused on using specific data sources to improve the cold-start recommendations for a specific application, in immersive recommendation, we developed techniques that are able to simultaneously profile multi-context data and improve recommendations in multiple applications beyond the cold-start phase and throughout a user's lifetime as the interests change.

Our profiling algorithm builds on the previous models for comparative text mining [34,51], but we additionally introduce a unique background topic [12] for each corpus to simultaneously learn the specific background noise for different contexts to improve the profile accuracy. Our recommendation model is an extension to Wang and Blei's CTM [11] and Agarwal and Chen's fLDA models [5] and belongs to the general framework of the regression-based latent factor [4]. However, most prior work in this line of research only considered categorical user attributes, such as gender or age [33], and thus cannot be directly applied to immersive recommendations, where dense high-dimensional user-generated data is available. In contrast, the proposed collaborative user-item regression model is able to utilize rich user data and significantly improve the recommendation performance.

7. CONCLUSION AND FUTURE WORK

We presented *immersive recommendation*, a user-centric recommendation model that empowers individuals to benefit from their diverse digital traces and enable a richer, more personally-relevant and desirable recommendation results. We proposed a topic-model-based algorithm to simultaneously profile multi-context user digital traces and a hybrid collaborative filtering model to improve the recommendation quality beyond the user-cold-start phase. We targeted news and local-event recommendations for their utility and societal importance and conducted a large-scale offline evaluation with Medium.com and Meetup.com users based on their publicly-available Twitter traces. We further verified the results with a 33-person user study with more diverse digital traces and the direct user evaluation. In almost every case, immersive recommendation showed significant improvement over the prior algorithms, which establishes the feasibility and justifies the potential efficacy of this new recommendation model.

While we only focus on text data in the present work, the promising results suggest a fruitful research avenue for the investigation of other digital traces, including location and travel histories, personally-created or viewed images, and the online purchase and consumption histories. An important potential benefit of immersive recommendation is to turn a recommendation system into a tool for awareness and aspiration. For example, the techniques proposed in the present work can be used to create a pathway for user interaction with the personalization/recommendation system to intentionally bias the system toward not only the user's observed behaviors, but the user's aspirational goals. Future work will explore how users can inform recommendation systems with their intentions so as to break out of their behavioral loops.

8. ACKNOWLEDGMENTS

We would like to thank Lucky Gunasekara, Arnaud Sahuguet, and the anonymous reviewers for insightful comments. This research is partly funded by AOL through the Connected Experiences Laboratory. The work is further supported by the small data lab at Cornell Tech, which receives funding from UnitedHealth Group, Google, Pfizer, RWJF, NIH and NSF.

9. REFERENCES

[1] Mozilla, firefox interest dashboard. `https://www.mozilla.org/en-US/firefox/interest-dashboard/`, 2014.

[2] Context-Aware LDA. `https://github.com/changun/CA-LDA`, 2015.

[3] F. Abel, N. Henze, E. Herder, and D. Krause. Interweaving public user profiles on the web. In *User Modeling, Adaptation, and Personalization*, pages 16–27. Springer, 2010.

[4] D. Agarwal and B.-C. Chen. Regression-based latent factor models. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 19–28, 2009.

[5] D. Agarwal and B.-C. Chen. fLDA: matrix factorization through latent dirichlet allocation. In *Proceedings of the third ACM International Conference on Web Search and Data Mining*, pages 91–100, 2010.

[6] A. Agresti. *Analysis of ordinal categorical data*, volume 656. John Wiley & Sons, 2010.

[7] E. M. Airoldi, D. M. Blei, E. A. Erosheva, and S. E. Fienberg. *Handbook of Mixed Membership Models and Their Applications*. CRC Press, 2014.

[8] X. Amatriain, J. M. Pujol, and N. Oliver. I like it... i like it not: Evaluating user ratings noise in recommender systems. In *User Modeling, Adaptation, and Personalization*, pages 247–258. Springer, 2009.

[9] T. Bansal, M. Das, and C. Bhattacharyya. Content driven user profiling for comment-worthy recommendations of news and blog articles. In *Proceedings of the 9th ACM Conference on Recommender Systems*, pages 195–202. ACM, 2015.

[10] P. Bhattacharya, M. B. Zafar, N. Ganguly, S. Ghosh, and K. P. Gummadi. Inferring user interests in the twitter social network. In *Proceedings of the 8th ACM Conference on Recommender Systems*, RecSys '14, pages 357–360, 2014.

[11] D. M. Blei. Probabilistic topic models. *Communications of the ACM*, 55(4):77–84, Apr. 2012.

[12] C. Chemudugunta and P. S. M. Steyvers. Modeling general and specific aspects of documents with a probabilistic topic model. In *Proceedings of the Conference on Advances in Neural Information Processing Systems*, volume 19, page 241, 2007.

[13] J. Cohen. *Statistical power analysis for the behavioral sciences*. Academic press, 2013.

[14] S. Cooper. The difference between living in new york and san francisco. `https://medium.com/@sarahcpr/the-difference-between-living-in-new-york-and-san-francisco-3e8ae58832a5`, 2015. Accessed: 2015-04-21.

[15] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (methodological)*, pages 1–38, 1977.

[16] D. A. Dillman, J. D. Smyth, and L. Melani. *Internet, mail, and mixed-mode surveys: the tailored design method*. Wiley & Sons Toronto, 2011.

[17] D. Estrin. Small Data, Where N = Me. *Communications of the ACM*, 57(4):32–34, Apr. 2014.

[18] R. Forrest and A. Kearns. Social cohesion, social capital and the neighbourhood. *Urban studies*, 38(12):2125–2143, 2001.

[19] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, 22(1):5–53, 2004.

[20] M. Hofer. Adolescents' development of individual interests: A product of multiple goal regulation? *Educational Psychologist*, 45(3):149–166, 2010.

[21] B. Hogan. The presentation of self in the age of social media: Distinguishing performances and exhibitions online. *Bulletin of Science, Technology & Society*, 2010.

[22] Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. In *The Eighth IEEE International Conference on Data Mining*, pages 263–272, 2008.

[23] S. Iyengar and D. R. Kinder. *News that matters: Television and American opinion*. University of Chicago Press, 2010.

[24] G. Keppel. *Design and analysis: A researcher's handbook.* Prentice-Hall, Inc, 1991.

[25] B. Klimt and Y. Yang. The Enron corpus: A new dataset for email classification research. In *Machine learning: ECML 2004*, pages 217–226. Springer, 2004.

[26] Q. V. Le and T. Mikolov. Distributed representations of sentences and documents. *arXiv preprint arXiv:1405.4053*, 2014.

[27] R. Likert. A technique for the measurement of attitudes. *Archives of psychology*, 1932.

[28] X. Lu, C. Wang, J.-M. Yang, Y. Pang, and L. Zhang. Photo2trip: generating travel routes from geo-tagged photos for trip planning. In *Proceedings of the International Conference on Multimedia*, pages 143–152, 2010.

[29] H. Ma, H. Yang, M. R. Lyu, and I. King. Sorec: social recommendation using probabilistic matrix factorization. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 931–940, 2008.

[30] A. E. Marwick and danah boyd. I tweet honestly, I tweet passionately: Twitter users, context collapse, and the imagined audience. *New media & society*, 13(1):114–133, 2011.

[31] A. K. McCallum. {MALLET: A Machine Learning for Language Toolkit}. 2002.

[32] A. Mnih and R. Salakhutdinov. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems*, pages 1257–1264, 2007.

[33] S.-T. Park and W. Chu. Pairwise preference regression for cold-start recommendation. In *Proceedings of the third ACM conference on Recommender systems*, pages 21–28, 2009.

[34] M. Paul and R. Girju. Cross-cultural analysis of blogs and forums with mixed-collection topic models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3*, pages 1408–1417, 2009.

[35] M. Pennacchiotti, F. Silvestri, H. Vahabi, and R. Venturini. Making your interests follow you on twitter. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, pages 165–174. ACM, 2012.

[36] R. D. Putnam. Bowling alone: America's declining social capital. *Journal of Democracy*, 6(1):65–78, 1995.

[37] R. Řehřek and P. Sojka. Software framework for topic modelling with large corpora. 2010.

[38] S. Robertson and H. Zaragoza. *The probabilistic relevance framework: BM25 and beyond*. Now Publishers Inc, 2009.

[39] S. Sedhain, S. Sanner, D. Braziunas, L. Xie, and J. Christensen. Social collaborative filtering for cold-start recommendations. In *Proceedings of the 8th ACM Conference on Recommender Systems*, pages 345–348, 2014.

[40] G. Shani and A. Gunawardana. Evaluating recommendation systems. In *Recommender systems handbook*, pages 257–297. Springer, 2011.

[41] Y. Shi, A. Karatzoglou, L. Baltrunas, M. Larson, N. Oliver, and A. Hanjalic. Climf: learning to maximize reciprocal rank with collaborative less-is-more filtering. In *Proceedings of the Sixth ACM Conference on Recommender Systems*, pages 139–146, 2012.

[42] Y. Shi, M. Larson, and A. Hanjalic. Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges. *ACM Computing Surveys*, 47(1):3, 2014.

[43] R. Sinha and K. Swearingen. The role of transparency in recommender systems. In *CHI '02 Extended Abstracts on Human Factors in Computing Systems*, pages 830–831, 2002.

[44] C. Wang and D. M. Blei. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 448–456, 2011.

[45] M. Weimer, A. Karatzoglou, Q. V. Le, and A. Smola. Maximum margin matrix factorization for collaborative ranking. *Advances in Neural Information Processing Systems*, 2007.

[46] M. Yan, J. Sang, and C. Xu. Mining cross-network association for youtube video promotion. In *Proceedings of the ACM International Conference on Multimedia*, pages 557–566, 2014.

[47] L. Yang, C.-K. Hsieh, and D. Estrin. Beyond classification: Latent user interests profiling from visual contents analysis. *arXiv preprint arXiv:1512.06785*, 2015.

[48] S.-H. Yang, B. Long, A. Smola, N. Sadagopan, Z. Zheng, and H. Zha. Like like alike: joint friendship and interest propagation in social networks. In *Proceedings of the 20th International Conference on World Wide Web*, pages 537–546, 2011.

[49] X. Yang, Y. Li, and J. Luo. Pinterest board recommendation for twitter users. In *Proceedings of the 23rd ACM International Conference on Multimedia*, pages 963–966, 2015.

[50] X. Yu, X. Ren, Y. Sun, Q. Gu, B. Sturt, U. Khandelwal, B. Norick, and J. Han. Personalized entity recommendation: A heterogeneous information network approach. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, pages 283–292, 2014.

[51] C. Zhai, A. Velivelli, and B. Yu. A cross-collection mixture model for comparative text mining. In *Proceedings of the tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 743–748, 2004.

[52] Z. Zhao, Z. Cheng, L. Hong, and E. H. Chi. Improving user topic interest profiles by behavior factorization. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1406–1416, 2015.

[53] C. Zhong, D. Karamshuk, and N. Sastry. Predicting pinterest: Automating a distributed human computation. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1417–1426, 2015.