

Scalable Distributed Inference of Dynamic User Interests for Behavioral Targeting

Amr Ahmed, Yucheng Low
School of Computer Science, CMU
Carnegie Mellon University
Pittsburgh, PA 15213, USA
{amahmed,ylow}@cs.cmu.edu

Mohamed Aly, Vanja Josifovski,
Alexander J. Smola
Yahoo! Research, Santa Clara, CA 95051, USA
{aly, vanjaj, smola}@yahoo-inc.com

ABSTRACT

Historical user activity is key for building user profiles to predict the user behavior and affinities in many web applications such as targeting of online advertising, content personalization and social recommendations. User profiles are temporal, and changes in a user's activity patterns are particularly useful for improved prediction and recommendation. For instance, an increased interest in car-related web pages may well suggest that the user might be shopping for a new vehicle. In this paper we present a comprehensive statistical framework for user profiling based on topic models which is able to capture such effects in a fully *unsupervised* fashion. Our method models topical interests of a user dynamically where both the user association with the topics and the topics themselves are allowed to vary over time, thus ensuring that the profiles remain current.

We describe a streaming, distributed inference algorithm which is able to handle tens of millions of users. Our results show that our model contributes towards improved behavioral targeting of display advertising relative to baseline models that do not incorporate topical and/or temporal dependencies. As a side-effect our model yields human-understandable results which can be used in an intuitive fashion by advertisers.

Categories and Subject Descriptors

I.2.6 [ARTIFICIAL INTELLIGENCE]: Learning; G.3 [PROBABILITY AND STATISTICS]:

General Terms

Algorithms, Experimentation

Keywords

Computational Advertising, Distributed Inference, Large-scale, Online Inference, User Modeling

1. INTRODUCTION

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

KDD'11, August 21–24, 2011, San Diego, California, USA.
Copyright 2011 ACM 978-1-4503-0813-7/11/08 ...\$10.00.

Computational advertising, content targeting, personalized recommendation, and web search, all benefit from a detailed knowledge of the interests of the user in order to personalize the results and improve relevance. For this purpose, user activity is tracked by publishers and third parties through browser cookies that uniquely identify the user visits to web sites. A variety of different actions are associated with each user as web page visits, search queries, etc. They are distilled into a compact description of the *user profile*.

In this paper we focus on generation of compact and effective user profiles from the history of user actions. One of the key challenges in this task is that the user history is a mixture of user interests over a period of time. In order to reason about the user, the personalization layer needs to be able to separate between the distinct interests to avoid degrading the user experience.

The framework presented in this paper is based on topic models. It is able to capture the *user interests* in an *unsupervised* fashion. We demonstrate the effectiveness of this framework for the *audience selection* task in display advertising. Audience selection is one of the core activities in display advertising where the goal is to select a good target for a particular campaign.

There are several challenges with topical analysis of the user action streams. First, for a production strength system, the analysis has to scale to hundreds of millions of users with tens, if not hundreds of actions daily. Most of the topical analysis models are computationally expensive and cannot perform at this scale. The second issue is that of time dependence: Users' interests change over time and it is this change that proves to be commercially very valuable since it may indicate purchase intents. For example, if a user suddenly increases the number of queries for the Caribbean and cruises, this may indicate interest in a cruise vacation. The appearance of this new topic of interest in the user's stream of actions can be predictive of his interest. Furthermore, there are external effects that may govern user behavior. For instance, an underwater oil spill does not convert users into deep sea exploration aficionados yet it affects search behavior and thus needs to be filtered out.

Finally, generating good features that *describe* user behavior does not necessarily translate into good features that are *predictive* of commercially relevant decisions. In other words, we ultimately want to obtain discriminatively useful features. Hence we would like to obtain a user profiling algorithm which has at least *the potential* of being adapted to the profiling task at hand.

We propose a coherent approach using Bayesian statis-

tics to achieve all those goals and we show that it excels at the task of predicting user responses for display advertising targeting. In summary our contributions are as follows:

- We develop a Bayesian approach for online modeling of user interests.
- Our model incrementally adapts both the user-topic associations and the topic composition based on the stream of user actions.
- We show how topic models can be used for improved targeting for display advertising, an application requiring analysis of *tens of millions* of users in real time.

2. FRAMEWORK AND BACKGROUND

We begin by spelling out the intuition behind our model in qualitative terms and describe the concepts behind the statistical model capable of addressing core profile generation issues, which we will address below and in Section 3:

- unsupervised topic acquisition
- dynamic topic descriptions
- dynamic user topical description
- filtering-out of global events

When categorizing user behavior it is tempting to group users into categories. That is, one might aim to group users together based on their (similar) behavior. While this has been used successfully in the analysis of user data [1] it tends to be rather limited when it comes to large numbers of users and large amounts of behavioral data: With millions of users it makes sense to use more than 1000 clusters in order to describe the user behavior. However, an inflation of clusters decreases the interpretability of the clusters considerably. Secondly, we would like to exploit sharing of statistical strength between clusters, e.g. if a number of clusters contain users interested in Latin music one would like to transfer modeling knowledge between them.

This can be addressed by topic models such as Latent Dirichlet Allocation (LDA) [6]. There objects (users) are characterized by a mixture of topics (interests). It is intuitively clear that such a topical mixture may carry considerably more information. For instance, even if we are allowed to choose only two out of 1000 topics we can already model 500,000 possible combinations. In other words, we can model the same variety of observations in a considerably more compressed fashion than what would be needed if we wanted to describe things in terms of clusters. The other key advantage of such models is that they are fully *unsupervised*. That is, they require *no* editorial data. Instead, they build the entire model by attempting to describe the data available. A key benefit in this setting is that it allows us to cover the entire range of interests rather than only those that an editor might be aware of. As a result this method is not language and culture specific. In the following we will be using the words topic and interest interchangeably.

Our strategy is thus to describe users as a mixture of topics and to assume that each of their actions is motivated by choosing a topic of interest first and subsequently a word to describe that action from the catalog of words consistent with that particular interest. For the purpose of this paper the **user actions** are either to issue a query or view an object (page, search result, or a video). We represent each user as a *bag of words* extracted from those actions and we use the term *user action* to denote generating a **word** from

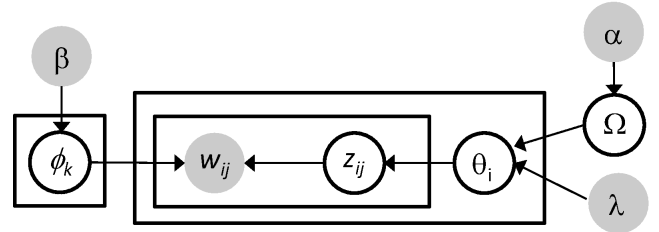


Figure 1: Latent Dirichlet Allocation.

this bag. For instance, when issuing a query, each word in the query is an action. Similarly, when viewing a video, each tag of the video represents an action, and when viewing a page, each word in the title of the page is an action.

In the following subsections we first review Latent Dirichlet Allocation (LDA) and then give an equivalent Polya-Urn scheme representation which will serve as the basis for the time-varying model described later in the paper.

2.1 Latent Dirichlet Allocation

LDA was introduced by [6] for the purpose of document modeling. The associated graphical model is given in Figure 1. In it, the variables w_{ij} correspond to the j -th action of user i that we observe. All remaining variables are considered latent, i.e. they are unobserved.

The associated graphical model of LDA assumes a static setting in which the order of the actions of each user w_{ij} is irrelevant. θ_i represents user i 's specific distribution over topics (interests), and ϕ_k represents the topic distribution over words. Typically one chooses for $p(\theta_i|\lambda\Omega)$ and for $p(\phi_k|\beta)$ Dirichlet distributions, hence the name Dirichlet Allocation. The full generative process is specified as follow:

1. Draw once $\Omega|\alpha \sim \text{Dir}(\alpha/K)$.
2. Draw each topic $\phi_k|\beta \sim \text{Dir}(\beta)$.
3. For each user i :
 - (a) Draw topic proportions $\theta_i|\lambda, \Omega \sim \text{Dir}(\lambda\Omega)$.
 - (b) For each word
 - (a) Draw a topic $z_{ij}|\theta_i \sim \text{Mult}(\theta_i)$.
 - (b) Draw a word $w_{ij}|z_{ij}, \phi \sim \text{Multi}(\phi_{z_{ij}})$.

K denotes the number of topics. The above generative process factors out the global distribution over topics into two parts: Ω and λ . Ω is a normalized vector, i.e., a probability distribution and represents the prior distribution over topics (interests) across users, and is sampled from a symmetric Dirichlet prior parametrized by α/K . λ on the other hand, is a scalar, and controls how each user's distribution over interests might vary from the prior distribution. This factored representation is critical for the rest of this paper.

2.2 A Polya-Urn Representation of LDA

In order to make explicit how the above LDA generative process captures local and global users' interests, we present an equivalent representation obtained by integrating out the global topic distribution Ω and the user-specific topic distribution θ . Let us assume that user i has generated $j-1$ actions and considering generating action j . Among those $j-1$ actions, let n_{ik} represent the number of actions expressing interest k . To generate action j the user might choose to either repeat an old interest with probability proportional to n_{ik} (and increment n_{ik}) or to consider a totally new interest with probability proportional to λ . In the former case, the user action is controlled by the user's mindset, and in

the latter case the new interest is decided by considering the global frequency of interests across all users. We let m_k represent the global frequency by which interest k is expressed across users. Thus, user i can select to express interest k with probability proportional to $m_k + \alpha/K$ and increment m_k as well as n_{ik} . This allows the model to capture the fact that not every action expressed by the user represents a genuine user’s interest. For example, a user might search for articles about the oil spill just because of the large buzz created about this incident. In summary we have

$$P(z_{ij} = k | w_{ij} = w, \text{rest}) \propto \left(n_{ik} + \lambda \frac{m_k + \frac{\alpha}{K}}{\sum_{k'} m_{k'} + \alpha} \right) P(w_{ij} | \phi_k) \quad (1)$$

In the literature, this model is known as the hierarchical Polya-Urn model [5] in which previously expressed interests have a *reinforcing* effect of being re-expressed either at the user-level or across users. Moreover, this model is also equivalent to a *fixed-dimensional* version of the hierarchical Dirichlet process (HDP) [25]. In Figure 2 we graphically show this representation. The static model is equivalent to a single vertical slice (with no prior over m nor n). This figure makes explicit that every visit to the global process by user i creates a new table which is denoted by a big circle. Thus m_k represents the total number of tables associated with topic k across all users. Note that in the standard HDP metaphor, to generate an action j , one first selects a table proportional to its popularity (or a new table with probability $\propto \lambda$), and then generates the action from the topic associated with the selected table. The process described above is strictly equivalent since the probability of choosing topic k under this standard HDP metaphor is thus equal to the number of words assigned to all tables serving topic k (which we denoted by n_{ik}). In Section 3, we will describe the time-varying version of this process.

3. TIME-VARYING USER MODEL

We now introduce our model — the Time-Varying User Model (TVUM). In Section 2.1, we assumed that user actions are fully exchangeable, and that user’s interests are fixed over time. It is reasonable though to assume that a user’s interests are not fixed over time, instead, we may assume that the proportions change and that new interests may arise. For instance, after the birth of a baby users will naturally be interested in parenting issues and their preferences in automobiles may change. Furthermore, specific projects, such as planning a holiday, purchasing a car, obtaining a mortgage, etc. will lead to a marked change in user activity. In the classical document context of LDA this is the equivalent of allowing the topics contained in a document to drift slowly as the document progresses. In the context of users this means that we assume that the daily user interest distribution is allowed to drift slowly over time.

We divide user actions into epochs based on the time stamp of the action. The epoch length depends on the nature of the application and can range from a few minutes to a full day. Figure 2 depicts the generative process of TVUM. User actions inside each epoch are modeled using an epoch-specific, fixed-dimensional hierarchical Polya-Urn model as described in Section 2.2. As time goes by, three aspects of the static model change: the global distribution over interests, the user-specific distribution over interests

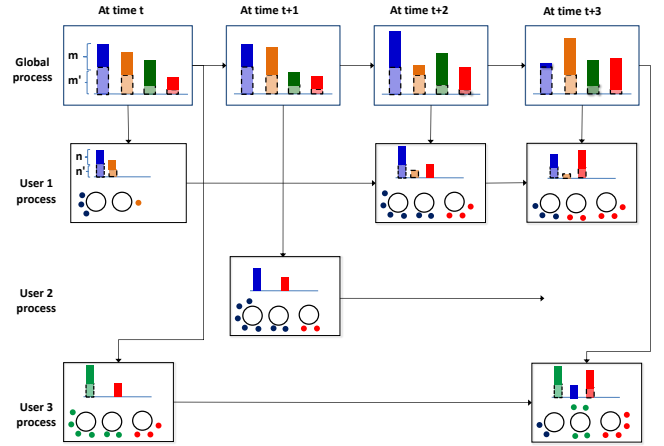


Figure 2: A time-varying hierarchical Polya-urn representation of the TVUM process (can also be regarded as a fixed-dimensional, fully-evolving recurrent Chinese restaurant franchise process [3]). Here (unlike in [3]) all levels are evolving. The top level represents the global process that captures the global topic trends. Each row gives the process for a given user. Each bar represents a topic’s popularity (either global popularity m , or user-specific popularity n). The dotted (bottom) part of each bar represents the prior (\tilde{m} for global topic prior, and \tilde{n} for user-specific topic prior). To avoid clutter, we only show first-order dependencies, however, the global process evolves according to an exponential kernel and the user-specific processes evolve according to a multi-scale kernel as shown in Figure 3. Note here also that user interactions are sparse and users need not appear at all days nor enter in the system in the same day. Finally, small circles in user processes represent words according to topics.

and the topic distribution over words ϕ . We address each of which in the following subsections and then summarize the full process in Section 3.4.

3.1 Dynamic Global Interests

The global trend of interests change over time. For example, the release of the iPhone 4 results in an increase in the global interest of the ‘mobile gadgets’ topic (intent) for a few days or weeks to follow. To capture that, we use a similar idea to that introduced in [3] and stipulate that the popularity of an interest k at time t depends both on the topic usages at time t , m_k^t , and on its historic usages at previous epochs, where the contribution of the previous epochs is summarized as \tilde{m}_k^t . We use exponential decay with kernel parameter κ defined as follows:

$$\tilde{m}_k^t = \sum_{h=1}^{t-1} \exp\left(-\frac{h-t}{\kappa}\right) m_k^h. \quad (2)$$

3.2 Dynamic User’s Interests

Now we turn to model the dynamics in the user-specific interests. The topic trends of a given user n_{ik} is now made dependent on time via n_{ik}^t . This is after all, the very variable that we wish to estimate. We could use the same exponential decay idea that we used in modeling the change in the global trends of interests, however, the change in the user’s interests over time is rather more complicated. Consider the set of actions observed from user i up to time $t - 1$ as depicted in Figure 3. For simplicity, assume that all of these

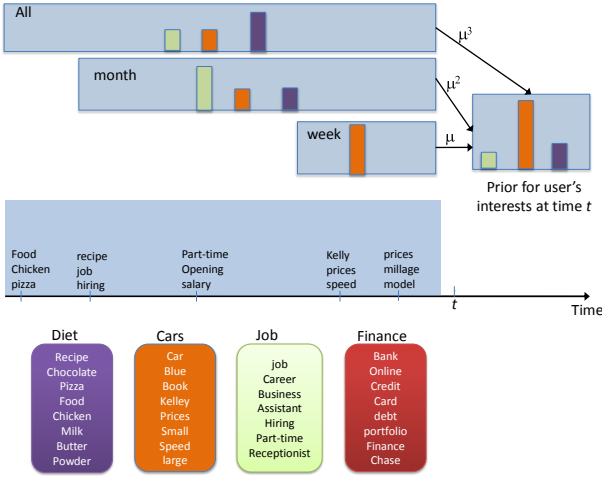


Figure 3: An illustration describing how TVUM captures users' long and short-term interests.

actions are words from queries. Each day, we observe a set of queries issued by the user. In the figure we also list a set of 4 interests (topics). For each topic, we give the top words in its distribution as recorded in ϕ . What could be the expected interests of the user at time t ? To answer this question, we observe that we can factor out the interests of the user into short-term interests and long-term (or persistent) interests. For example, this user has a long-term interest in Diet (Food) related materials. From the user history we can also discern two transient short-term interests localized on time: one over finding a job and the other over buying a car. To discover the long-term interests of the user, we could count the frequency of expressing interest k across the whole user history. This gives the long-term interests of the user. As depicted in Figure 3, this shows that the user has a long-term interest over Diet. Similarly we could compute the same quantity for the recent week and month to get the short-term interests of the user. As shown in Figure 3, it is clear that in the recent month, the user was mainly looking for a job and then started to look for a car in the recent week. Thus to get the *expected* user-specific's popularity over interests $\tilde{\mathbf{n}}_i^t$ for user i at time t , we combine these three levels of abstractions using a weighted sum

$$\tilde{\mathbf{n}}_i^t = \mu_{\text{week}} \tilde{\mathbf{n}}_i^{t,\text{week}} + \mu_{\text{month}} \tilde{\mathbf{n}}_i^{t,\text{month}} + \mu_{\text{all}} \tilde{\mathbf{n}}_i^{t,\text{all}}. \quad (3)$$

Here $\tilde{\mathbf{n}}_i^t$ is our estimate for user i 's interest over topic k at time t , and $\tilde{\mathbf{n}}_i^{t,\text{week}} = \sum_{h=t-7}^t n_{ik}^h$, n_{ik}^h is the frequency of expressing interest k by user i at time h , and $\tilde{\mathbf{n}}_i^{t,\text{week}}$ is just the sum of these frequencies over the past week. The other two quantities $\tilde{\mathbf{n}}_i^{t,\text{all}}$ and $\tilde{\mathbf{n}}_i^{t,\text{month}}$ are defined accordingly over their respective time ranges. The set of weights μ gives the contribution of every abstraction to the final estimate. In Figure 3, we show the final estimate at time t and it shows that we expect that the user will continue to look for cars, might still look for 'diet' related content, or with some probability considers a new job. There are a few observations here. First, if we had used an exponential decay as we employed for modeling the global trends, the model would have forgotten quickly about the user's long-term interest in 'diet'. This happens because the user history is sparse and the time between the user's activities might range from days to weeks. However, for modeling the change in the global trends of topics, it is enough to consider the preceding days

to get a good estimate about the future: if an interest starts to fade, it is likely that it will die soon unless the data at future days tells us something else. The second observation is that $\tilde{\mathbf{n}}_i^t$ is not enough to capture what the user will look for at time t , as the user might still consider a new interest. For example, the user might develop an interest in obtaining a loan to finance his new car purchase. To model this effect, we combine the aforementioned prior with the global distribution over interests, as we will detail in Section 3.4, to generate the user actions at time t .

Finally, it is hard to fit or tune three different weights as in (3), thus we use $\mu_{\text{week}} = \mu$, $\mu_{\text{month}} = \mu^2$ and $\mu_{\text{all}} = \mu^3$, where $\mu \in [0, 1]$. For values of μ close to 0, more weight is given to the short-term interests and for values of μ close to 1 a uniform weight is given to all levels which implies more weight given to the long-term interests as they are aggregates over a longer time period. μ could be either optimized per user or set to a default value for all users.

3.3 Dynamic Topics

The final ingredient in our model is rendering the topics themselves dynamic. Indeed as we observe more user interactions, we expect the topic distribution over words to change in response to world events like the release of a new car. This feature is related to earlier work in dynamic topic models [7, 3] which uses non-conjugate logistic normal priors (which are challenging during inference) and [16] which uses multi-scale priors similar to Figure 3. Our approach here resembles [16] but with a simpler form for the prior. We achieve this dynamic effect by making the topic distribution over words ϕ_k time dependent, i.e. we have ϕ_k^t which now depends on the smoothing prior $\tilde{\beta}_k^t$ in addition to the static prior β . The component $\tilde{\beta}_{kw}^t$ of $\tilde{\beta}_k^t$ depends on a decayed version of the frequencies of observing word w from topic k at times $1 : t - 1$. This is similar to the way we modeled the dynamics of the global distribution over interests, and we use an exponential decay kernel parametrized by κ_0 :

$$\tilde{\beta}_{kw}^t = \sum_{h=1}^{t-1} \exp\left(\frac{h-t}{\kappa_0}\right) n_{kw}^h, \quad (4)$$

Here n_{kw}^h is the frequency of observing word w from topic k at day h . Finally, we have $\phi_k^t \sim \text{Dir}(\tilde{\beta}_k^t + \beta)$

Note that the decision to *add* β to the above prior enforces that the components of the prior corresponding to new words (i.e. words that have $\tilde{\beta}_{kw}^t = 0$) is nonzero and as such new words can appear as time goes by.

3.4 Model Summary

We now put all the pieces together and give the full generative process of TVUM. Consider generating action j for user i at time t . User i can choose an interest k with probability proportional to $n_{ik}^t + \tilde{\mathbf{n}}_i^t$ and then increments n_{ik}^t . Alternatively, the user can choose a new interest with probability proportional to λ . To select this new interest, he considers the global trend of interests across users: he select interest k with probability proportional to $m_k^t + \tilde{m}_k^t + \alpha/K$ and increment m_k^t as well as n_{ik}^t . Finally the user generates the word w_{ij}^t from topic k 's distribution at time t , ϕ_k^t . Putting everything together, we have

$$P(z_{ij}^t = k | w_{ij}^t = w, \text{rest}) \propto \left(n_{ik}^t + \tilde{\mathbf{n}}_i^t + \lambda \frac{m_k^t + \tilde{m}_k^t + \frac{\alpha}{K}}{\sum_{k'} m_{k'}^t + \tilde{m}_{k'}^t + \alpha} \right) P(w_{ij}^t | \phi_k^t) \quad (5)$$

In fact, taking the limit of this model as $K \rightarrow \infty$ we get the recurrent Chinese restaurant franchise process as in [3] albeit with all levels being evolved and with different level-specific time-kernels. In [3] only the top level evolves since a given document does not persist across time epochs. Moreover, our fixed-dimensional approximation is more amenable for distributed inference and is still not highly affected by the number of topics as we will demonstrate in the experiments.

3.5 Inference

The problem of inferring interests for millions of users over several months is formidable and it considerably exceeds the scale of published work on scalable topic models including [23]. As a result exact inference, even by sampling, is computationally infeasible: for instance, in a collapsed sampler along the lines of [13, 26] the temporal interdependence of topic assignments would require inordinate amounts of computation even when resampling single topic assignments for user actions: it affects all actions within a range of one month if we use a correspondingly long dependence model to describe the topic and interest evolution.

A possible solution is to use Sequential Monte Carlo (SMC) methods such as those proposed by [9] for online inference of topic models. The problem is that due to the long-range dependence the particles in the SMC estimator quickly become too heavy, so we need to rebalance and resample fairly frequently. This problem is exacerbated by the sheer amount of data — we need to distribute the inference algorithm over several computers. This means that whenever we resample particles we need to transfer and update the state of many computers. Such an approach very quickly becomes infeasible and we need to resort to further approximations.

We make the design choice of essentially following an inference procedure. That is, we only perform forward inference through the set of dependent variables for each time step and we attempt to infer \mathbf{z}^t only given estimates and observations for $t' \leq t$. This allows us to obtain improving results as time progresses, alas at the expense of rather suboptimal estimates at the beginning of the inference chain.

A second reason to use forward sampling is practicality: data keeps on arriving at the estimator and we obviously would like to update the user profiles as we go along. This is only feasible if we need not revisit old data constantly while new data keeps on arriving. Our setting allows effectively for an online sampling procedure where we track the current interest distribution and the nature of interests relative to incoming data. Thus to summarize, our inference procedure incrementally runs a *fast* batch MCMC over the data at epoch t given the state of the sampler at earlier epochs.

Sampling Details

Our sampler resembles the collapsed, direct-assignment sampler with augmented representation as described in [25]. We collapse the topic multinomials (ϕ^t) and compute the posterior over \mathbf{z}^t given assignments to hidden variables at previous epochs. As noted in (5), sampling z would couple topic indicators across all users via \mathbf{m}^t : recall from Figure 2 that m_k^t is the number of tables across all users serving topic k . This fact is undesirable especially when users are distributed across machines. To remedy this, we augment the representation by instantiating and sampling the global topic distribution at time time, i.e. Ω^t . Thus the sampler alternate between sampling \mathbf{z}^t , \mathbf{m}^t , and Ω^t . It should be noted that

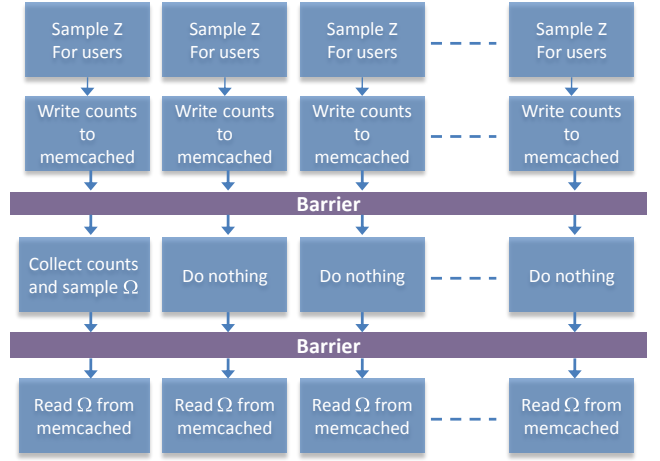


Figure 4: Synchronization during a sampling cycle.

collapsing ϕ^t also introduces coupling across users, however we deal with that using the architecture of [23].

Sampling z_{ij}^t : The conditional probability for z_{ij}^t has the same structure as in LDA albeit with the previously defined compound prior. We have,

$$P(z_{ij}^t = k | w_{ij}^t = w, \Omega^t, \tilde{\mathbf{n}}_i^t) \propto \left(n_{ik}^{t,-j} + \tilde{n}_{ik}^t + \Lambda \Omega^t \right) \frac{n_{kw}^{t,-j} + \tilde{\beta}_{kw}^t + \beta}{\sum_l n_{kl}^{t,-j} + \tilde{\beta}_{kl}^t + \beta} \quad (6)$$

where n_{kw}^t is the number of times a word w was sampled from topic k (this is known as the topic-word matrix). Λ denotes the influence of the common activity distribution. The superscript $-j$ means excluding the contribution of word j . The structure of (6) allows us to utilize the efficient sparse sampler described in [26] which in fact was another reason to our choice of using the augmented representation, i.e. for instantiating Ω^t .

Sampling m_k^t : Since our sampler does not explicitly maintain the word-table assignments, we need to sample $m_k^t = \sum_i m_{ik}^t$. Where m_{ik}^t is the number of tables in user i 's process serving topic k . This last quantity m_{ik}^t follows what is called the Antoniak distribution [25]. A sample from this distribution can be obtained as follows. First, set $m_{ik}^t = 0$, then for $j = 1 \dots n_{ik}^t$, flip a coin with bias $\frac{\lambda \Omega_k^t}{j-1 + \lambda \Omega_k^t}$, and increment m_{ik}^t if the coin turns head. The final value of m_{ik}^t is a sample from the Antoniak distribution. Recall that n_{ik}^t is the number of words expressing interest k from user i at time t .

Sampling Ω^t : By examining the structure of (1) and (5) and the equivalence between the construction in Section 2.2 and LDA, it is straightforward to see that

$$P(\Omega^t | \mathbf{m}^t, \tilde{\mathbf{m}}^t) \sim \text{Dir}(\tilde{\mathbf{m}}^t + \mathbf{m}^t + \alpha/K).$$

This constitutes a **single** Gibbs cycle. For each time epoch (day) we iterate this cycle **100 times** over all active users in that day. 100 iterations were enough due to 1) the small size of user observations at each day, and 2) the informative prior from earlier days.

Systems Aspects

To be able to apply the model to tens of millions of users, we use a distributed implementation following the architecture in [23]. The state of the sampler comprises the topic-word

set	days	users	vocabulary	campaigns	size
1	56	13.34M	100K	241	242GB
2	44	33.5M	100K	216	435GB

Table 1: Basic statistics of the data used.

counts matrix, and the user-topic counts matrix. The former is shared across users and is maintained in a distributed hash table using `memcached` [23]. The later is user-specific and can be maintained locally in each client. We distribute the users at time t across N clients. Each client executes a foreground thread to implement (6) and a background thread that synchronizes its local copy of the topic-word counts matrix with the global copy in `memcached`. As shown in Figure 4, after this step, Ω^t and m_k^t should be sampled. However, sampling Ω^t requires a *reduction* step across clients to collect and sum m_{ik}^t . First, each client writes to `memcached` its contribution for m_{ik}^t (which is the sum of the values of m_{ik}^t for users i assigned to this client), then the clients reach a *barrier* where they wait for each other to proceed to the next stage of the cycle. We implemented a sense-reversing barrier algorithm which arranges the nodes in a tree and thus has a latency that scales logarithmically with N [21]. After this barrier, all counts are in `memcached` and as such one client sums these counts to obtain \mathbf{m}^t , uses it to sample Ω^t , and finally writes the sampled value of Ω^t back to the `memcached`. All other clients wait for this event (signaled via a barrier), and then read the new value of Ω^t and this finalizes a single cycle.

4. EXPERIMENTS

- We show that the discovered user interests are effective when used as features for behavioral targeting. This holds both in the sense of generating *interpretable* features and in the sense of generating *predictive* models.
- Secondly we demonstrate the scalability of our inference algorithm across both the number of machines and number of users. In particular we demonstrate that our algorithm is capable of processing tens of millions of users. This is over 10 times larger than the datasets analyzed by [23] who only discuss LDA.
- Third, we show that our model and inference algorithm are robust to a wide range of parameter settings. Hence it applies to a wide range of problems without the need to excessively hand-tune its parameters.

4.1 Datasets and Tasks

We used two datasets each of which spans approximately two months. The exact dates of each dataset are hidden for privacy reasons. In each case we collected a set of ad-campaigns and sampled a set of users who were shown ads in this campaign during the covered time period. For each user, we represented the users' interaction history as a sequence of tokens. More precisely, the user *is represented* at day t as a *bag of words* comprising the queries they issued at this day and the textual *content* of the pages they view on this day. This constitutes the input to all the models discussed in the following subsections. Thus all datasets are Yahoo! privacy policy compliant, since we don't retain the precise page the user viewed on Yahoo! Sports (for instance) but rather retain only the description of the page at a higher level in terms of words extracted from the content.

For evaluation purposes, we also collect the user's response to individual ads: the ads they converted on (positive response); and the ads they ignored either by not clicking or

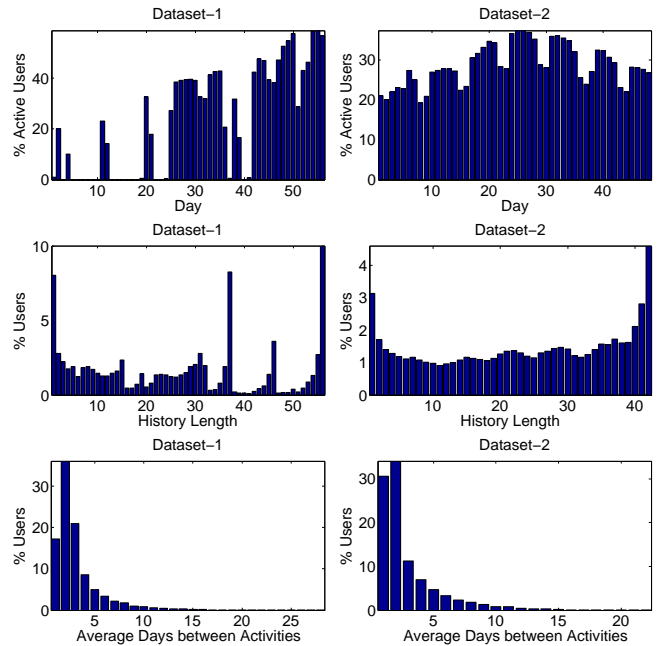


Figure 5: Characteristics of the data sets.

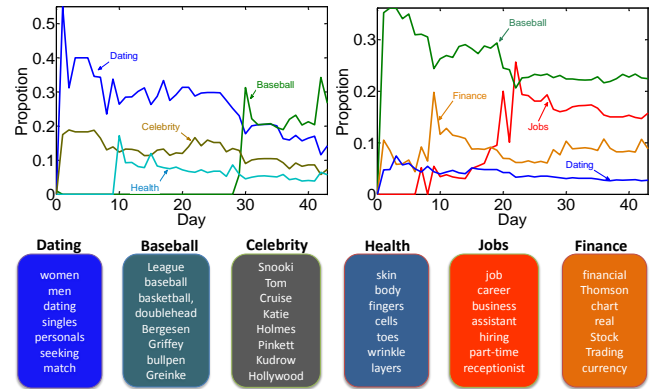


Figure 6: Dynamic interests of two users.

not converting post-click (negative response). We used the users' responses to ads in the last week of the dataset as a test set and the remaining responses as a training set. The characteristics of each dataset are shown in Figure 5 and Table 1. As evident from Figure 5, the first dataset is sparser than the second which presents a challenge for any model.

Our task is to build a model based on users' history that can predict whether or not a user will convert given an ad. We break this problem in two parts. First, we translate the user's history into a feature-vector profile using the TVUM and other baselines. Second, the profiles are used to build an SVM model that is evaluated over the test set using the area under the ROC curve as a metric.

4.2 Interpretability

Figure 6 qualitatively illustrates our model. For each user we plot the interest distribution over 6 salient topics. For each topic we print the top words as recorded in ϕ . The top part of the figure shows the dynamic interest of two users (User A — left; User B — right) as a function of time. Our model discovers that A has a *long-term* interest in dating, health, and celebrity albeit to a varying degree.

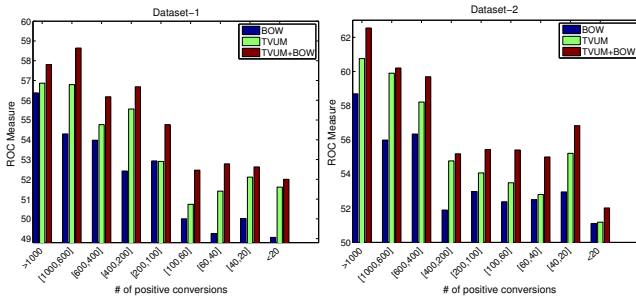


Figure 7: Performance vs. number of conversions.

In the last 10 days, A developed a *short-term* interest in baseball, quite possibly due to the global rise in interest in baseball (within our dataset). On the other hand B has a *long-term* interest in sports and finance; B was interested in dating in the first week, and was mainly looking for jobs in the last month. These discovered time-varying features for each user are useful for timely targeting of users with products. The following experiments buttress this intuition.

4.3 Performance

To show that the proposed TVUM is effective for extracting user interest we compare it to two baselines: a bag of words model (BOW) which does *not* incorporate any topical or temporal dependencies, and a **static LDA** model which incorporates topical information but *no* time dependence. The bag of words (BOW) model just outputs the concatenation of the user history up until time t as its representation of the user’s interest at time t . This is in fact a very strong baseline. One advantage of the TVUM is that it is capable of processing user data in an online fashion, that is, we may estimate the user interest distribution one day at a time, sweeping through the dataset.

The training data for the SVM consists of the user’s distribution over intent at time t (or a bag of words alternatively) and the user’s response to the ad. For the TVUM, we used the following default parameters:¹ we set the number of topics $K = 200$, $\beta = 0.01$, $\mu = e^{-1} \approx .6$, $\kappa = \kappa_0 = \lambda = 1$.

In Figure 7, we show a comparison between this BOW baseline and the TVUM as a function of the number of positive conversions in each campaign. In this figure, we also show a third model that combines the BOW features with the TVUM features. As evident, our model improves over the baseline in both datasets across all the campaigns. We observe that, in general, the number of positive examples in ad campaigns is scarce. Therefore, models that feature a compact representation of the user’s history overall perform better and can leverage the BOW features to further improve its performance. We can also notice from this figure that the gain over the BOW baseline is larger in campaigns with medium to small number of conversions. We also experimented with a variation of the BOW baseline featuring **BOW+time decay**. In this variation of BOW, we use an *exponential decay* of the form $a^{t'-t}$ over the *word counts* in the BOW baseline such that distant history will be given less weight at time t , however, the best result in this case was obtained when $a = 1$ (i.e. ignoring time), thus our model was able to leverage temporal information in a better way.

¹It is possible to learn μ for users with long history (greater than two weeks). The likelihood of μ is Dirichlet-multinomial compound and μ can be learnt using gradient descent. We leave this as a future work.

Table 2: Average ROC measure (TVUM: time-varying user model, BOW: bag of words baseline).

	BOW	TVUM	TVUM+BOW	LDA+BOW
dataset 1	54.40	55.78	56.94	55.80
dataset 2	57.03	58.70	60.38	58.54

Table 3: Time (in minutes) per a Gibbs iteration

topics		50	100	200
dataset 1	50 machines	0.7	0.9	1.2
dataset 2	100 machines	0.9	1.1	1.5

Table 4: Effect of # topics on predictive accuracy.

	topics	TVUM	TVUM + BOW
dataset 1	50	55.32	56.01
	100	55.5	56.56
	200	55.8	56.94
dataset 2	50	59.10	60.40
	100	59.14	60.60
	200	58.7	60.38

Table 5: The effect of the topic-decay parameter on the performance of TVUM+BOW.

Decay(κ_0)		none	1	2	3
dataset 1	200 topics	56.2	56.94	56.51	56.51
dataset 2	100 topics	60.68	60.60	60.40	60.40

To summarize the performance of each model over all campaigns, we give in Table 2 a weighted average of the ROC measure across campaigns, where the weight of each campaign is proportional to the number of positive examples in the campaign.

Finally, Table 2 also shows a comparison with a static LDA model+BOW features. Due to the large scale of this dataset, we can not apply batch LDA directly over it. Instead, we sampled 2M users from the training set and built a static LDA model over them. Then, we used variational inference to infer our estimate of the user’s interest at time t , where t can be in the training or test set. Note here that LDA does not have the notion of a user in the model in a sense that when predicting the user’s history at time t_1 and later at t_2 , the user history up until time t_1 and time t_2 are presented as separate documents to the model. As evident from Table 2, our model beats static LDA on both datasets due to its ability to model time, and to maintain a coherent representation of the user as a time-evolving entity (note that in this application of predicting conversion, an increase of 1-2 points of the ROC measure is considered significant)

4.4 Speed and Efficiency

One key advantage of our model is its ability to process data in an online and distributed fashion. This allows us to process data at a scale quite unlike any system that we are aware of. Rather than processing a small number of millions of documents (i.e. users) [27, 28] while requiring a large cluster of computers, and rather than being able to process tens of millions of documents without time dependence [23], we are able to process tens to hundreds of millions of users in a dynamic fashion at a speed of **two hours per day** of data being processed. This is quite a significant improvement and makes topic models suitable for Yahoo’s user base.

Table 3 provides the time for a single Gibbs-sampling iter-

ation for different number of topics and using different number of machines. Since dataset 2 is almost double the size of dataset 1 in terms of the number of users, we doubled the number of machines when processing dataset 2. As shown in this table, the time per iteration is almost the same across the two datasets. The slight increase in time for dataset 2 arises because of the cost of synchronization and inter-client communications which was kept low thanks to the logarithmic complexity of the tree-based barrier algorithm.

4.5 Sensitivity Analysis

We evaluate the results’ sensitivity to the model’s parameters. Bayesian models depend on the specification of the parameters which encode our prior belief. However, no matter what prior is used, as long as the prior does not exclude the correct answer, the model can reach the true conclusion as the size of the dataset increases (which is true in our application). We first study the effect of the number of topics and then the effect of the decay parameter κ_0 for the topic distribution over words. The effect of κ was negligible.

Number of Topics: We varied the number of topics in the range 50, 100, 200. Results are presented in Table 4. From these tables, we first observe that the effect on the final performance is not quite large in both datasets. This is largely due to the fact that our inference algorithm optimizes over the topic’s global distribution Ω which allows it to adapt the *effective* number of topics for each dataset [29]. For dataset 1, the performance increases as we add more topics. This is due to the sparsity and inhomogeneity of this dataset. Thus, the arrival of new users at each day requires more topics to model their intents properly. Whereas dataset 2 was more homogeneous, and as such, 100 topics were enough. We recommend setting the number of topics in the low hundreds for behavioral targeting applications.

Topic-Distribution Decay: Decaying the topic’s word distribution has two important roles. First, it allows the model to recover from a poor initial estimate, and second, it enables the model to capture drifts in the topic’s focus. As shown in Table 5 this feature helps in the first dataset due to the initial poor estimate of the topics in the first few days because of the low number of available users. However for the second homogeneous dataset, the performance *slightly* increases if we turn this feature off. We recommend setting κ_0 to a value between 1 and 3. It is possible to learn this parameter, however, this requires the storage of all the past topic distributions over time which is prohibitive in terms of storage requirements. In our implementation, because of the form of the decay function, we only need to discount the topic-word counts at the end of each day and use this as the initial estimate for the following day.

5. RELATED WORK

The emergence of the web has allowed for collection and processing of user data magnitudes larger than previously possible. This has resulted in a spike of interest in user data analysis and profile generation as reported in [10, 12, 17, 19]. Profile generation has been reported for a few different applications. In [24] the authors describe profiles for search personalization. Here, the authors build profiles based on the query stream of the user and users similar to it. The authors also report an alternative that is based on the relevance feedback approaches in information retrieval over the

documents that the user have perceived as relevant. Both techniques are orthogonal to the work presented in this paper and could be used to produce a potentially richer set of features that will serve as an input to the topical analysis.

User profile generation is also studied in other online settings and also for content recommendation (e.g., [14, 18, 20]). Most of these focus on detecting the user’s short term vs long term interest and using these in the proposed application. In our case, we blend the short term and long term interests into a single profile. A survey of user profile generation can be found in [12].

In the area of audience selection, Provost et al. [22] have recently shown that user profiles can be built based on co-visitation patterns of social network pages. These profiles are used to predict the performance of brand display advertisements over 15 campaigns. In [10] the authors discuss prediction of clicks and impressions of events (queries, page views) within a set of predefined categories. Supervised linear poisson regression is used to model these counts based on a labeled set of user-class pairs of data instances.

While there are many profile generating algorithms satisfying a partial set of requirements outlined in this paper we are unaware of methods covering the entire range of the desiderata. In the topical analysis area, there are many predictive algorithms which try modeling the observed data via static generative model. The examples include singular value decomposition of the (user, action) matrix thus yielding a technique also known as Latent Semantic Indexing [11] or more advanced techniques such as Probabilistic Latent Semantic Indexing [15] or Latent Dirichlet Allocation [6]. However, while reducing the dimensionality of the space (the number for unique features), LSI and PLSI yield a dense vectorial representation. All of these approaches are static in terms of the user and to be able to apply new data we need to recompute the topical model from scratch. Finally, several models exist in the literature that could accommodate the evolution of topic global trends, topic distribution over words [7, 16] and the number of topics [3]. However, we are not aware of *any* attempts to model the intra-document drift of topics which corresponds to a user in our application – which is the *main contribution* of this paper.

6. CONCLUSION

In this paper, we addressed the problem of user profile generation for behavioral targeting. We presented a time-varying hierarchical user model that captures both the user’s long term and short term interests. While models of this nature were not known to scale for web-scale applications, we showed a streaming distributed inference algorithm that both scales to tens of millions of users and adapts the inferred user’s interest as it gets to know more about the user. Our learnt user representation was experimentally proven to be effective in computational advertising.

There are several directions for future research. First, we focused in this paper on the dynamic aspect of user profiles, and integrated all available user actions into a single vocabulary. While this approach seems promising, it would be beneficial to model each facet of the user action independently perhaps using a different language model for each facet. It is also possible to build a supervised model as in [8] that uses the users’ responses to historical ads in inferring the user’s interests.

7. REFERENCES

- [1] D. Agarwal and S. Merugu. Predictive discrete latent factor models for large scale dyadic data. *KDD*, 2007.
- [2] A. Ahmed and E. P. Xing. Dynamic non-parametric mixture models and the recurrent chinese restaurant process In *SDM*, pages 219–230. SIAM, 2008.
- [3] A. Ahmed and E. P. Xing. Timeline: A dynamic hierarchical dirichlet process model for recovering birth / death and evolution of topics in text stream. In *UAI*, 2010.
- [4] A. Asuncion, P. Smyth, and M. Welling. Asynchronous distributed learning of topic models. In *NIPS*, pages 81–88. MIT Press, 2008.
- [5] D. Blackwell and J. MacQueen. Ferguson distributions via polya urn schemes. *The Annals of Statistics*, 1973.
- [6] D. Blei, A. Ng, and M. Jordan. Latent Dirichlet allocation. *JMLR*, 3:993–1022, 2003.
- [7] D. M. Blei and J. D. Lafferty. Dynamic topic models. In *ICML*, volume 148, pages 113–120. ACM, 2006.
- [8] D. M. Blei and J. D. McAuliffe. Supervised topic models. In *NIPS*. MIT Press, 2007.
- [9] K. R. Canini, L. Shi, and T. L. Griffiths. Online inference of topics with latent dirichlet allocation. In *AISTATS*, 2009.
- [10] Y. Chen, D. Pavlov, and J. F. Canny. Large-scale behavioral targeting. In *KDD*, pages 209–218, 2009.
- [11] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Am. Soc. for Information Science*, 41, 1990.
- [12] S. Gauch, M. Speretta, A. Chandramouli, and A. Micarelli. User profiles for personalized information access. In *LNCS 4321*, Springer, 2007.
- [13] T.L. Griffiths and M. Steyvers. Finding scientific topics. *PNAS*, 101:5228–5235, 2004.
- [14] A. Hassan, R. Jones, and K. L. Klinkner. Beyond DCG: User behavior as a predictor of a successful search. In *WSDM 2010*, pages 221–230, 2010.
- [15] T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 2001.
- [16] T. Iwata, T. Yamada, Y. Sakurai, and N. Ueda. Online multiscale dynamic topic models. In *KDD*, 2010.
- [17] H. R. Kim and P. K. Chan. Learning implicit user interest hierarchy for context in personalization. In *IUI*, 2003.
- [18] R. Kumar and A. Tomkins. A characterization of online search behavior. In *WWW*, 561–570, 2010.
- [19] L. Li, Z. Yang, B. Wang, and M. Kitsuregawa. Dynamic adaptation strategies for long-term and short-term user profile to personalize search. In *ADWM*, 2007.
- [20] L. Li, W. Chu, J. Langford, and R. Schapire. A contextual bandit approach to personalized news article recommendation. In *WWW*, 661–670, 2010.
- [21] J. Mellor-Crummey and M. L. Scott. Algorithms for scalable synchronization on shared-memory multiprocessors. *ACM TOCS*, 9(1):21–65, February 1991.
- [22] F. J. Provost, B. Dalessandro, R. Hook, X. Zhang, and A. Murray. Audience selection for on-line brand advertising: privacy-friendly social network targeting. In *KDD*, pages 707–716, 2009.
- [23] A.J. Smola and S. Narayanamurthy. An architecture for parallel topic models. In *VLDB*, 2010.
- [24] K. Sugiyama, K. Hatano, and M. Yoshikawa. Adaptive web search based on user profile constructed without any effort from users. In *WWW*, pages 675–684, 2004.
- [25] Y. Teh, M. Jordan, M. Beal, and D. Blei. Hierarchical dirichlet processes. *JASA*, 2006.
- [26] L. Yao, D. Mimno, and A. McCallum. Efficient methods for topic model inference on streaming document collections. In *KDD'09*, 2009.
- [27] Y. Wang, H. Bai, M. Stanton, W. Chen, and E. Chang. PLDA: Parallel latent dirichlet allocation for large-scale applications. In *Proc. of 5th International Conference on Algorithmic Aspects in Information and Management*, 2009.
- [28] D. Newman, A. Asuncion, P. Smyth and M. Welling. Distributed Algorithms for Topic Models. In *Journal of Machine Learning Research*, 2009.
- [29] H. Wallach, D. Mimno and A. McCallum. Rethinking LDA: Why Priors Matter. In *Advances in Neural Information Processing Systems 22*, 2009.