Topic Modeling for Personalized Recommendation of Volatile Items

Maks Ovsjanikov[1] and Ye Chen[2]

[1] Stanford University
maks@stanford.edu
[2] Microsoft Corporation
yec@microsoft.com

Abstract. One of the major strengths of probabilistic topic modeling is the ability to reveal hidden relations via the analysis of co-occurrence patterns on dyadic observations, such as document-term pairs. However, in many practical settings, the extreme sparsity and volatility of co-occurrence patterns within the data, when the majority of terms appear in a single document, limits the applicability of topic models. In this paper, we propose an efficient topic modeling framework in the presence of volatile dyadic observations when direct topic modeling is infeasible. We show both theoretically and empirically that often-available unstructured and semantically-rich meta-data can serve as a link between dyadic sets, and can allow accurate and efficient inference. Our approach is general and can work with most latent variable models, which rely on stable dyadic data, such as pLSI, LDA, and GaP. Using transactional data from a major e-commerce site, we demonstrate the effectiveness as well as the applicability of our method in a personalized recommendation system for volatile items. Our experiments show that the proposed learning method outperforms the traditional LDA by capturing more persistent relations between dyadic sets of wide and practical significance.

1 Introduction

Probabilistic topic models have emerged as a natural, statistically sound method for inferring hidden semantic relations between terms in large collections of documents, e.g., [4,12,11]. Most topic-based models start by assuming that each term in a given document is generated from a hidden topic, and a document can be characterized as a probability distribution over the set of topics. Thus, learning high level semantic relations between documents and terms can be reduced to learning the topic models from a large corpus of documents. At the core of many learning methods for topic-based models lies the idea that terms that often occur together are likely to be explained by the same topic. This is a powerful idea that has been successfully applied in a wide range of fields including computer vision, e.g., [9], and shape retrieval [14].

One of the limitations of direct topic modeling, however, is that co-occurrence patterns can be very sparse and are often volatile. For example, terms cannot

J.L. Balcázar et al. (Eds.): ECML PKDD 2010, Part II, LNAI 6322, pp. 483–498, 2010.

be readily substituted by images and documents by websites in the LDA model, since the vast majority of images will only occur in a single website. Nevertheless, it is often possible to obtain large amounts of *unstructured meta-data*, annotating the input. In this paper, we show that accurate topic modeling can still be performed in these settings, and occurrence probabilities can be computed despite extreme sparsity of the input data.

Our motivating application is a personalized recommendation system of volatile items, which aims to exploit past behavior of users to estimate their preferences for a large collection of items. Our data consists of eBay buyers and their behavioral history. The set of items at eBay is large, constantly evolving (over 9 million items added every day as of August 2009), and each item is only loosely categorized by eBay sellers, while obtaining an accurate catalog for all items is a very difficult task. Therefore, the overlap between purchase and even browsing history of individual users is minimal, which greatly complicates obtaining an accurate topic model for users and items [6].

In this paper we address these challenges by leveraging the unstructured meta-data that accompanies the user-item interactions. Specifically, we map both users and items to a common latent topic space by analyzing the search queries issued by users, and then obtain user and item profiles using statistical inference on the latent model in real time. Thus, the search queries act as a link between users and items, and, remarkably, allow us to build a useful latent model without considering direct user-item interaction. We also show how to use this decomposition to efficiently update occurrence probabilities when new items or users arrive.

2 Related Work

Our work lies on the crossroads between probabilistic topic modeling and personalized recommendation (see e.g., [17] and [1] for surveys of these two fields). The primary objective of probabilistic topic models [4,12,11] and their variants, is to capture latent topical information from a large collection of discrete, often textual, data. These methods are very popular due to their relative simplicity, efficiency and the results which are often easy to interpret. However, most topic models require a relatively stable vocabulary and a significant overlap of terms across documents, which is often difficult to enforce in practice.

Personalized recommendation systems aim to recommend relevant items to users based primarily on their observed behavior, e.g., search personalization [15], Google News personalization [8], and Yahoo! behavioral targeting [7] among others. Most personalization techniques are supervised or semi-supervised, and fit the model parameters to some known user preference values. More recently, several techniques have been proposed to leverage available meta-data to build topic models, which are then used for recommendation, e.g., [2,18]. In most cases, however, users and items are coupled during topic modeling. In our setting, this can lead to overfitting in the presence of sparse data and inefficiencies if the model parameters need to be recomputed whenever a new user or item enter the system. Our method overcomes these issues by decoupling users and items when

building the topic model. This not only allows us to obtain more stable topic models, but much more importantly allows us to model new users and items in real time.

Finally, our personalized recommendation system is similar, in spirit, to recently proposed Polylingual Topic Models [16], that aim to learn a consistent topic model for a set of related documents in different languages. In this work, the lack of overlap of terms across documents in different languages is countered by the fact that similar documents will have similar topic distributions, and thus the topic model can be inferred consistently across languages. This technique, however, requires pre-clustering of documents into similar sets, whereas our goal is to infer *personalized* topic models for a set of independent users without a clear cluster structure.

3 Topic Models with Triadic Observations

Although our approach is general, for clarity of exposition we concentrate on inferring unobserved user preferences for a large set of highly volatile items. In this section we describe the setting in detail, and give an overview of our method.

3.1 Motivation and Overview

Our primary goal is to infer the probability that a user u is interested in some item i. In the text setting, this is similar to obtaining the probability that a document will contain a given term. However, the set of items is extremely large and volatile, implying that the overlap between individual users' item browsing or purchasing histories is minimal. This prohibits standard user modeling as well as collaborative filtering techniques based either on finding similar users or factorizing the user-item matrix. Note, however, that in addition to the direct user-item interactions, in the majority of cases, users issue search queries to arrive at the desired items. This means that a single user-item observation can be augmented to a user-query-item triple. This augmentation only exacerbates the sparsity issue, however, since the set of triples is at least as sparse as the original user-item matrix. Therefore, we project this tensor onto the subspace of users and queries. In this space, topic modeling can be performed using standard techniques. Furthermore, since each item is equipped with the set of queries that were used to retrieve it, we use statistical inference on the set of item-query pairs to learn the probability that a given item is characterized by each latent topic. Finally, since the users and items are now modeled in a common reduced latent topic space, the user-item preferences can be obtained by combining the user and item distributions over the latent topics. Intuitively, the search queries serve as a link, which allows us to model both users and items in a common latent topic space. Our approach is therefore applicable whenever there exist meta-data drawn from a relatively stable vocabulary which allows linking two possibly heterogeneous and ephemerally coupled sets.

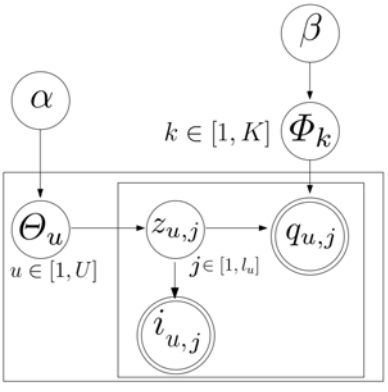


Fig. 1. Graphical representation of our Generative Model

Note that our approach can work with most state-of-the-art latent variable models (e.g., pLSI [12], LDA [4] or GaP [5]). We describe it using the structure of LDA, since it is a fully generative model, natural for our extension.

3.2 Generative Model

Our generative model starts by assuming a hidden set of user intentions that are responsible for generating both queries and items. In other words, a user searching for an item is guided by a certain interest, characterized by the latent topic, which results in a search query and an observed item. Formally, let i, q, u and k represent an item, query, user, and latent topic respectively, then $P(q, i|u)$ is the probability of user u observing a query-item pair (q, i). The simplest way to capture this probability is by assuming conditional independence of items and queries given the underlying latent topic:

$$\begin{aligned} P(q, i|u) &= \sum_k P(q|k, i, u)P(k, i|u) \\ &= \sum_k P(q|k)P(i|k)P(k|u). \end{aligned} \tag{1}$$

Following LDA, we assume that $P(k|u)$ is a multinomial probability distribution Θ_u over a fixed set of K topics. We interpret each element k of Θ_u as the interest of user u in topic k. We also assume that $P(q|k)$ is a multinomial distribution over the set of V distinct *search queries*, which can be represented by a vector Φ_k. Note that unlike the standard unigram model in LDA, a search query can contain multiple words or special characters. This is crucial in our setting since in a web commerce site, queries often contain brand names, having multiple words with conceptually different meanings (e.g. "Banana Republic"). On the other hand, due to high specialization of the web commerce domain the size of the vocabulary of queries is comparable with the size as individual terms.

Thus dealing with queries directly avoids the otherwise lossy n-gram extraction. Similarly to LDA, we introduce Dirichlet priors on Θ_u and Φ_k with specified symmetric hyperparameters α and β respectively. See Figure 1 for a graphical representation of our generative model.

Given this generative model, the marginal probability of a user u generating a particular query q is:

$$P(q|u) = \sum_k P(k|u)P(q|k) = \sum_k \Theta_u(k)\Phi_k(q). \tag{2}$$

Thus, similarly to LDA, our generative model assumes that each search query $q_{u,j}$ by user u corresponds to a latent topic $z_{u,j}$, sampled independently according to $P(z_{u,j} = k) = \Theta_u(k)$. Then, the search query is drawn according to Φ_k, i.e., $P(q_{u,j} = q|z_{u,j} = k) = \Phi_k(q)$.

In the following, we interpret $P(q|u)$ as the user's preference for a query q. Ultimately, we are interested in the user's preference for an item $P(i|u)$. Note that a straightforward extension of LDA would entail learning the topic probabilities over items $P(i|k)$, and using them to estimate $P(i|u)$. This, however, would require jointly modeling user and item topic models. Instead, we integrate, over all users, the topic model with triadic observations as in Eq. (1):

$$\begin{aligned} P(q,i) &= \int_u \sum_k P(q|k)P(k,i|u)P(u)du \\ &= \sum_k P(q|k)P(k|i)P(i), \end{aligned} \tag{3}$$

where the probability $P(k|i)$ can be interpreted as the probability that the latent topic k is responsible for generating an instance of item i. We encode these probabilities by a multinomial distribution *over the set of topics*, or a vector $\Psi_i \in \mathbb{R}^k$. Following LDA, we impose a Dirichlet prior γ on Ψ_i.

We stress that modeling $P(k|i)$ rather than $P(i|k)$ is crucial in allowing our method to decouple user and item behavior as well as to efficiently learn the model parameters for new users and items without recomputing the topic model. This decoupling is only made possible by the presence of textual meta-data (search queries) that links the users and items.

Estimating the model parameters from a corpus of queries generated by a population of users can then be regarded as a Bayesian approach to collaborative filtering, since it leverages co-occurrence information from multiple users to learn user preferences for unseen data. We also discuss the relation of our method to traditional collaborative filtering in the next section.

4 Personalized Recommendation

Recall that the motivating application of our proposed learning method was to derive a statistical formulation for recommendation of volatile items. In this section, we carry out the derivation using the generative model described above.

We also establish a theoretical connection between our method and matrix factorization methods in collaborative filtering.

Intuitively, a generative model is well-suited for a recommendation system since it allows mimicking the user behavior after learning the model parameters. In other words, once the model parameters have been learned, one can generate queries as well as items for a particular user automatically by simply following the generative process. The key property that translates this observation into a recommendation system, is that probabilistic topic models allow to assign meaningful probabilities to unobserved pairs, through the hidden layer of topics. Thus, a recommendation system based on latent topic modeling is unlikely to result in the same item that the user has already observed. However, the recommended query and item will have a high probability of belonging to the latent topic that the user has shown preference for in the past.

4.1 Query Recommendation

Given the user preference vector for latent topics Θ_u and the latent topic distributions $\Phi_k, k \in 1..K$, we recommend search queries to the user u by simulating the generative process described in Section 3.2. Namely, to generate a query q, first sample a latent topic z with $P(z = k) = \Theta_u(k)$, and then pick a query q from Φ_z, s.t. $P(q = t) = \Phi_z(t)$. Note that unlike traditional recommendation systems, which suggest queries or items that the user is most likely interested in, this process is randomized. This stochastic delivery mechanism allows us to diversify the recommendations while maintaining relevancy to the user, and yield a 100% recall of the entire inventory in an asymptotic sense.

4.2 Item Recommendation

Note that the process described above also allows us to recommend items to users. To do this, we issue the recommended queries to the search engine and use the retrieved results as the recommended items. In this way, the queries used for recommendation will correspond to likely queries in the latent topic k, for which the user has shown interest. This means, in particular, that the queries used for recommendation will rarely come from the set of queries issued by the user in question. Instead, they will be the most representative queries in the user's topics of interest. Note that this recommendation step is possible only because we perform topic modeling on full search queries, rather than individual terms, since reconstructing a query from a set of possibly unrelated search terms is a difficult task.

In real time to recommend N items, we can generate N distinct search queries, by first picking N latent topics from Θ_u *without replacement*, and use the top item provided by the search engine for each of these queries. This allows us to significantly diversify the recommended set while keeping it relevant to the user.

The preference $P(i|u)$ of user u for item i is given by:

$$P(i|u) = \sum_k P(i|k)P(k|u) = \sum_k \frac{P(i)}{P(k)} P(k|i)P(k|u)$$

Note that a symmetric Dirichlet prior, with parameters $\alpha_i = \alpha_j \ \forall \ i, j$, is invariant to permutations of the variables. In other words: $p(\Theta) = p(\Pi\Theta)$, for any permutation matrix Π. Moreover, if $\Theta' = \Pi\Theta$, the Jacobian determinant of this transformation is always 1, so $d\Theta' = d\Theta$, and the marginal:

$$P(k) = \int_{\Theta} \Theta_k p(\Theta) d\Theta = \int_{\Theta'} \Theta'_j p(\Theta') d\Theta' = P(j) \ \forall \ k, j$$

Thus, under symmetric Dirichlet prior on Θ_u, the marginal distribution $P(k)$ of latent topics is uniform and can be factored out, meaning $P(i|u) \propto P(i)\Theta_u^T \Psi_i$, where $P(i)$ is the likelihood of item i.

4.3 Relation to Latent Factor Models

We also point out that the generative model presented above is similar in spirit, to latent factor models in collaborative filtering, e.g., [13,3]. These methods map each user and each item to points in a common Euclidean space, where the prediction for user u's preference for item i is approximated by: $\hat{r}_{ui} = p_u^T q_i$, with a possible addition of a baseline predictor [13].

Note that in our model, $\hat{r}_{ui} \propto P(i)\Theta_u^T \Psi_i$, and therefore, traditional latent factor models can be thought of imposing a uniform prior on the space of items. Also note that unlike standard latent factor models, our topic model is derived via the relations between users and search queries, and is independent of user-item interactions. As we show in the next section, this allows us to efficiently infer model parameters for new users and items and perform recommendation.

5 Learning and Inference

In this section we describe the learning and inference procedures for our generative model. As mentioned earlier, our emphasis is on decoupling user and item interactions. Moreover, since the number of items exceeds the number of users by orders of magnitude, we aim to first learn the latent topic parameters by considering user behavior, and then adopt a fast inference procedure to obtain the topic distributions for each item.

5.1 Model Fitting

The marginal distribution of users and search queries described in Section 3.2 mirrors the generative model for documents and terms in LDA. Therefore, estimating Φ_k from data, can be done in a similar way as for LDA. Here, we adopt the Gibbs sampling approach [11], which has been shown to yield accurate results efficiently in practice.

The input to our user-based model estimation is a set of users, with a list of search queries that each user issued in a fixed period of time, where repetitions are naturally allowed. The goal of the Gibbs sampler is to determine for each

query in the dataset the latent topic that this query was generated by. Then, the model parameters Φ_k can be computed as statistics on these topic assignments. The main idea of Gibbs sampling for LDA is to derive the distribution:

$$p_k = P(z_{u,j} = k \mid \boldsymbol{z}_{\neg u,j}, \boldsymbol{u}, \boldsymbol{q}), \tag{4}$$

where $z_{u,j}$ is the topic responsible for generating query j of user u, $\boldsymbol{z}_{\neg u,j}$ are topic assignments for all other queries, while $\boldsymbol{u}$ and $\boldsymbol{q}$ are users and queries respectively. Once this distribution is established, we run the Gibbs sampler with a random initialization of $\boldsymbol{z}$ until convergence. In our model, as in LDA:

$$\begin{aligned} p_k =& \frac{P(\boldsymbol{z}, \boldsymbol{u}, \boldsymbol{q})}{P(\boldsymbol{z}_{\neg u,j}, \boldsymbol{u}, \boldsymbol{q})} \\ =& \frac{P(\boldsymbol{q}|\boldsymbol{z})}{P(q_{u,j})P(\boldsymbol{q}_{\neg u,j}|\boldsymbol{z}_{\neg u,j})} \frac{P(\boldsymbol{z}|\boldsymbol{u})}{P(\boldsymbol{z}_{\neg u,j}|\boldsymbol{u})}. \end{aligned} \tag{5}$$

Both $P(\boldsymbol{q}|\boldsymbol{z})$ and $P(\boldsymbol{z}|\boldsymbol{u})$ can easily be derived by integrating over the parameter space, since e.g.:

$$\begin{aligned} P(\boldsymbol{q}|\boldsymbol{z}) &= \int_{\boldsymbol{\Phi}} P(\boldsymbol{q}|\boldsymbol{z}, \boldsymbol{\Phi}) P(\boldsymbol{\Phi}|\beta) d\boldsymbol{\Phi} \\ &= \prod_{k=1}^{K} \frac{1}{\mathrm{B}(\beta)} \mathrm{B}(\boldsymbol{z}^k + \beta), \end{aligned} \tag{6}$$

where $\boldsymbol{z}^k(q)$ is the number of times query q is assigned to topic k in the topic assignment vector $\boldsymbol{z}$ and B is the multivariate beta function. The final distribution for the Gibbs sampler is:

$$p_k \propto \frac{\boldsymbol{z}^k_{\neg u,j}(q_{u,j}) + \beta}{\sum_{w=1}^{V} \left(\boldsymbol{z}^k_{\neg u,j}(w) + \beta\right)} \frac{\boldsymbol{z}^k_{\neg u,j}(u) + \alpha}{\sum_{j=1}^{K} \left(\boldsymbol{z}^j_{\neg u,j}(u) + \alpha\right)}, \tag{7}$$

where $\boldsymbol{z}^k_{\neg u,j}(q)$ is the number of times query q is assigned to topic k in $\boldsymbol{z}_{\neg u,j}$, and $\boldsymbol{z}^k_{\neg u,j}(u)$ is the number of times topic k is assigned to a query by user u. Once the Gibbs sampler converges to a stationary distribution, the model parameters can be computed as:

$$\Phi_k(q) = \frac{\boldsymbol{z}^k(q) + \beta}{\sum_{w=1}^{V} \boldsymbol{z}^k(w) + \beta}. \tag{8}$$

We approximate $P(i)$ by measuring the fraction of times that item i was observed in the corpus of user-item pairs.

5.2 Inference

Once the model parameters Φ_k have been estimated, the inferential task consists of learning Θ_u and Ψ_i, for a user u or item i.

First, suppose the data set consists of a single user u with all the search queries issued by this user in a fixed time period, and our goal is to estimate the user preference vector Θ_u. The Gibbs sampling procedure described above allows us to perform inference in a straightforward fashion. Again, for every query $q_{u,j}$ in the new dataset, we aim to determine which latent topic z is responsible for generating this query. For this, we run the Gibbs sampler in the same fashion as above, while *only iterating over queries of user u*. The probability distribution for the Gibbs sampler is nearly identical to Eq. (7):

$$\begin{aligned} p_k \propto & \frac{\boldsymbol{n}^k(q_{u,j}) + \boldsymbol{z}^k_{\neg u,j}(q_{u,j}) + \beta}{\sum_q^V \left(\boldsymbol{n}^k(q_{u,j}) + \boldsymbol{z}^k_{\neg u,j}(q) + \beta\right)} \\ & \cdot \frac{\boldsymbol{z}^k_{\neg u,j}(u) + \alpha}{\sum_j^K \left(\boldsymbol{z}^j_{\neg u,j}(u) + \alpha\right)}, \end{aligned} \tag{9}$$

where $\boldsymbol{z}$ is the assignment of latent topics to queries *only of user u*, while $\boldsymbol{n}^k(q_{u,j})$ is the number of times query $q_{u,j}$ was assigned to topic k *in the model fitting step*. Note that $\boldsymbol{n}$ is the only part of the distribution that depends on the model fitting, and can be seen as a precomputed sparse $V \times K$ matrix of counts. After convergence, the user preference vector Θ_u is:

$$\Theta_u(k) = \frac{\boldsymbol{z}^k(u) + \alpha}{\sum_{j=1}^K \boldsymbol{z}^j(u) + \alpha}. \tag{10}$$

To derive the inferential procedure for topic probability vector Ψ_i, given an item i, we assume that the data consists of all queries used by all users to arrive at item i. Then, our goal is to determine $z_{i,j}$: the latent topic responsible for generating query j used to arrive at item i. Note that in this case, Eq. (5) can be written as:

$$\begin{aligned} p_k =& P(z_{i,j} = k | \boldsymbol{z}_{\neg i,j}, \boldsymbol{i}, \boldsymbol{q}) \\ =& \frac{P(\boldsymbol{q}|\boldsymbol{i}, \boldsymbol{z}) P(\boldsymbol{z}|\boldsymbol{i}) P(\boldsymbol{i})}{P(\boldsymbol{q}|\boldsymbol{i}, \boldsymbol{z}_{\neg i,j}) P(\boldsymbol{z}_{\neg i,j}|\boldsymbol{i}) P(\boldsymbol{i})} \\ =& \frac{P(\boldsymbol{q}|\boldsymbol{z}) P(\boldsymbol{z}|\boldsymbol{i})}{P(\boldsymbol{q}|\boldsymbol{z}_{\neg i,j}) P(\boldsymbol{z}_{\neg i,j}|\boldsymbol{i})}, \end{aligned} \tag{11}$$

where the third equality follows from the conditional independence of items $\boldsymbol{i}$ and queries $\boldsymbol{q}$ given $\boldsymbol{z}$ assumed in our generative model. Note that the final expression for p_k has the same form as Eq. (5). In addition, the Dirichlet prior γ assumed on Ψ_i, where $\Psi_i(k) = P(k|i)$, forces the distribution of topics given an item to have the same form as the distribution of topics given a user. Thus, we can run the Gibbs sampler in the identical way as for Θ_u, to get Ψ_i.

Note that if the user u issued N search queries (or the item i was observed through N search queries), one sampling iteration of the Gibbs sampler will require only N topic assignments. Convergence is usually achieved after several Gibbs iterations, and inference is very fast. Therefore, learning model parameters Θ_u and Ψ_i for new users and items through inference is very efficient and does not require extensive computations, unlike conventional matrix factorization models.

6 Experiments

Our dataset consists of the search queries entered by the eBay users over a two-month period in 2009. We only consider queries, for which the search engine returned at least one item in the "Clothing, Shoes & Accessories" (CSA) meta-category. Furthermore, during the model fitting step we remove casual users, who did not purchase any items in this time period. This greatly reduces the data set, and leaves 2.6M users, with an average of 2.5 queries per user per day. We also limit the vocabulary of queries to those entered by at least 30 users to reduce the complexity of the Gibbs sampler. This reduces the data by an additional 8 percent, so that our final vocabulary consists of approximately 50K search queries.

Figures 2(a) and 2(b) show two trends observed in the data. In particular, Figure 2(a) shows that over 50 percent of the users who bought items in the CSA meta-category of eBay during two months, only bought one item in this meta-category (while the average number of purchased items is 3, which is explained by the heavy tailed nature of the distribution). On the other hand, Figure 2(b) shows that the median number of subcategories (as defined by the current eBay taxonomy) that individual users looked at is 5, with the mean 10.7. This shows that users at eBay are willing to explore products in a variety of categories. In particular, this means that recommendations based purely on purchasing behavior may not be able to capture the wide range of interests that users have.

6.1 Choice of Dirichlet Priors α and β

One of the principal advantages of the LDA model over pLSI [12] is the flexibility of prior parameters α and β. Indeed, pLSI is a *maximum a posteriori* LDA model estimated with a uniform Dirichlet prior $\alpha = 1$ [10]. To demonstrate the

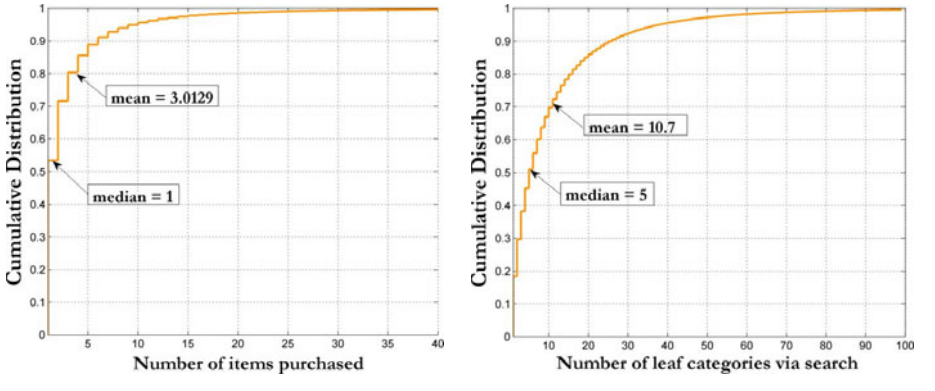
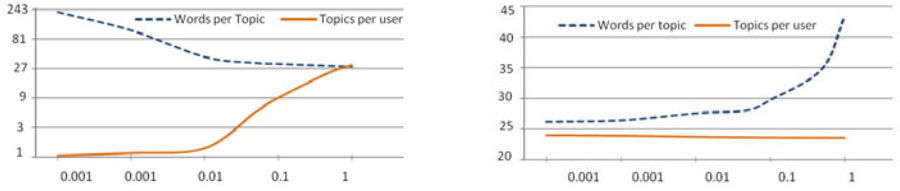


Fig. 2. (a) Number of items purchased by users (b) Number of subcategories of CSA explored by users



(a) Uniform prior α (b) Uniform prior β

Fig. 3. (a): Influence of α, at $\beta = 0.1$, (b): influence of β, at $\alpha = 0.5$

importance of this distinction, which holds even for a large dataset, we considered a sample of queries issued by 400K users, and estimated the model for 100 topics and various values of α and β. Figures 3(a) and 3(b) show the dependencies of the average number of topics per user and the average number of queries per topic on the priors α and β. For each user u we consider the median of Θ_u: the minimum number of topics, with cumulative distribution in Θ_u at least 0.5, and similarly, for each topic k, we consider the median of Φ_k.

Note that as α approaches 1, the average number of topics per user grows, which means that on average, users' interests become diffused over more topics. However, as a result of this, fewer queries are necessary to explain each topic, and Φ_k becomes more and more concentrated. The effect of β is less pronounced on the average number of topics per user, whereas the median number of queries per topic grows. This is consistent with the intuition that β controls how concentrated each topic is. Since our ultimate goal is to recommend items to users, we would like to have highly concentrated topics with the most relevant queries having a high probability. Furthermore, since the majority of users only explore a small number of subcategories, we expect each user to be interested in a small number of latent topics. Therefore, we use $\beta = 0.1, \alpha = 0.05$, so that on average, the median of Θ_u is 5.

6.2 Personalized versus Global Models

To evaluate the quality of our method, we compute the log-likelihood of unseen data given the model. For this, we first compute the model parameters Φ_k and Θ_u for the two-month data described above. This allows us to predict the preference $P(q|u)$ of user u for a particular query q using Eq. (2). The log-likelihood of a set of queries for a user is given simply as $\sum_j \log\left(P(q_j|u)\right)$. Thus, to evaluate the quality of our approach to personalized recommendation, we evaluate the log-likelihood of the search queries issued by the same users for which the model was estimated, but in the four days following the training period. A better predictive model would result in a smaller absolute value of the log-likelihood.

As a baseline predictor we use a global model, which is oblivious to individual user's preferences. This corresponds to setting the number of topics to 1, since in this case, under uniform Dirichlet priors, the probability $P(q|u)$ is simply the

Query	$\Phi_k(q)$
golf	0.2975
golf+shoes	0.0684
nike+golf	0.0652
tiger+woods	0.0571
callaway	0.0527
golf+shirts	0.0477
scotty+cameron	0.0375
greg+norman	0.0369
adidas+golf	0.0351
nike+golf+shirt	0.0339
golf+shirt	0.0334
puma+golf	0.0329
footjoy+classics	0.0299

(a)

Query	$\Phi_k(q)$
sunglasses	0.3168
ray+ban	0.1390
ray+ban+sunglasses	0.0888
mens+sunglasses	0.0459
mephisto	0.0388
fishing	0.0364
rayban	0.0341
aviator+sunglasses	0.0336
polarized+sunglasses	0.0331
ray+ban+aviator	0.0303
sun+glasses	0.0268
eyeglasses	0.0251
rayban+sunglasses	0.0245

(b)

Query	$\Phi_k(q)$
oakley	0.3337
oakley+sunglasses	0.2075
oakley+juliet	0.0561
oakley+gascan	0.0469
oakley+half+jacket	0.0405
oakley+m+frame	0.0387
oakley+radar	0.0342
oakley+frogskins	0.0293
oakley+flak+jacket	0.0281
oakley+rare	0.0206
oakley+display	0.0205
oakley+oil+rig	0.0197
oakley+romeo	0.0170

(c)

Query	$\Phi_k(q)$
gucci	0.2616
prada	0.1685
armani	0.1207
dolce+&+gabbana	0.0707
versace	0.0663
ferragamo	0.0651
dolce	0.0444
dolce+gabbana	0.0366
cavalli	0.0331
d&g	0.0171
roberto+cavalli	0.0169
bally	0.0155
dior	0.0142

(d)

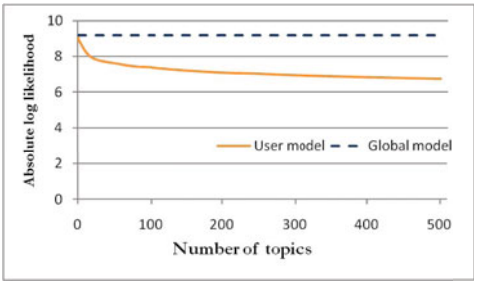
Fig. 4. Sample topics inferred using $\alpha = 0.05, \beta = 0.1$, and $K = 100$ along with queries having maximum probabilities $\Phi_k(q)$

fraction of times this query was used in the training data independent of u. In other words, each query is characterized by its *overall popularity* in the training data. Figure 5(a) shows the dependence of the absolute log-likelihood of the testing data on the number of topics K. Note that the global user-oblivious model with 1 topic results in 36% higher absolute log-likelihood than the model with 500 topics and 31% higher than the model with 200 topics. Interestingly, we do not observe any over-saturation phenomena, and the absolute log-likelihood decreases as far as 500 latent topics. However, the rate of the decrease slows down beyond 200 topics. Figure 5(b) shows that the improvement over the global model is significantly more pronounced for the users who issued many queries (in the training set). Thus, we achieve over 50 percent improvement in absolute log-likelihood using a model with 200 topics for users who entered over 400 search queries. This shows, not only that topical preferences are learned more accurately for users who have a longer search history, but also that the topics themselves represent persistent structure in the data.

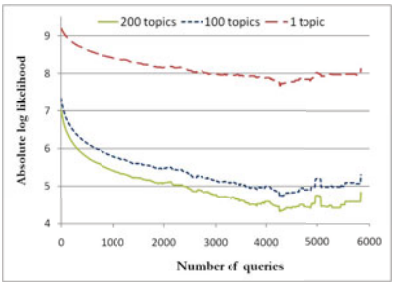
Figure 4 shows a few sample topics inferred using $K = 100$ together with queries having maximum probabilities $\Phi_k(q)$. Note the highly specific nature of each topic, which will allow us to achieve personalized recommendations. Thus, a user whose history contains many queries related to e.g. golf will be presented with a recommendation that corresponds to a likely query-item pair within this category (Figure 4(a)). Moreover, remark the flexibility provided by the topic modeling framework, where an individual topic can correspond to an activity (Figure 4(a)), a product (Figure 4(b)), a brand (Figure 4(c)), or even a set of conceptually related brands (Figure 4(d)). Finally, note that individual queries often contain multiple terms, and performing topic modeling on the query rather than term level is essential for accurate personalized recommendation.

6.3 Triadic versus Dyadic Models

Finally, we compare our method with standard LDA, to validate the advantage of capturing the hidden structure from a relatively persistent yet often-available intermediate layer, e.g., search queries in our case. Given a set of training data



(a) Absolute log-likelihood of the test data given as a function of number of topics.



(b) Absolute log-likelihood of the test data given as a function of number of queries.

Fig. 5. Absolute log-likelihood of the test data for different choices of K (a) and for users with different search history sizes (b)

in the form of triadic observations user-query-item, where each example triple means that a user issues a query and then conducts some transactions (click, watch, bid or purchase event) on an item, we first trained a topic model with triadic observations as described in Section 3. To leverage the semantically-rich query layer, the training set of triples was projected to a user-item matrix, while the item dimension uses queries (converting to corresponding items) as descriptors. In other words, the vocabulary consists of converting queries. On the other hand, a standard LDA was trained on a direct projection of user-item matrix by ignoring the query layer, where the vocabulary consists of all unique item ids. Now the difference in volatility between queries and item ids shall be appreciated, and this setting applies to many practical domains such as web images, graphic ads, and page URLs.

Let *word* w denote query q or item i for triadic and dyadic models respectively. After training and inference of both models, we have the posterior Dirichlet parameter given each user $P(k|u)$, and word multinomials conditioned on each topic $P(w|k)$. Given a test data set in the same triple form and collected after the training period, we then can compare the log likelihoods of user-item conversions under both models:

$$\ell = \sum_u \sum_w \log \left(\sum_k P(w|k)p(k|u) \right). \tag{12}$$

It is important to note that although w denotes different objects for different models, the log likelihood reflects the prediction quality against a same ground truth, which is the user-item conversions in future. Personalization is achieved by using different posterior per-user topical mixture $P(k|u)$ for each user. Online prediction can easily follow. Under triadic topic models, to predict and rank a candidate set of items, one needs to augment items with their historically converting queries.

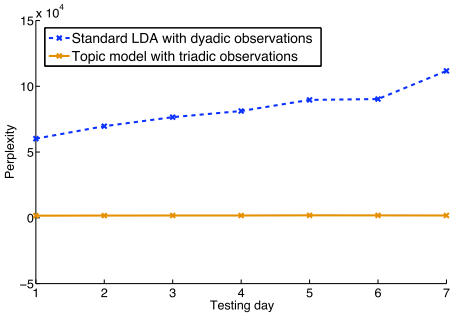


Fig. 6. Daily perplexity on one-week test data. Triadic model outperforms standard LDA, with the gap widening as the testing day is further away from the training day.

We trained both models using one-day worth of user-query-item conversion data, and evaluated both models against the following one-week test data on a daily basis. The training data contains 5.3M triples from 500K users, which gives a vocabulary size $V=$ 539,092 queries for the triadic model and $V=$ 2,186,472 item ids for the standard LDA. We set $\alpha = 0.05$, $\beta = 0.1$, and $K = 100$ as separately tuned in Section 6.1. For a normalized comparison, we report perplexity (two to the power of minus per-word test-set log likelihood in our case), as plotted in Figure 6. The results show that our proposed topic model with triadic observations consistently outperforms the standard LDA with dyadic observations, with the edge widening as testing day farther away from the training day. On day one right after the training date, the per-word log likelihood of the triadic model is -15.88, while a LDA yields -10.68. The over 30% log likelihood improvement translates to over 30 folds decrease in perplexity, with $PP^{\text{triadic}}_{\text{day 1}} = 1.6\text{K}$ and $PP^{\text{lda}}_{\text{day 1}} = 60\text{K}$. As the testing day moves farther away from the training day, the prediction performance of our approach stays stable, while that of a standard LDA deteriorates drastically. On day seven the perplexity gain of our approach over the conventional LDA becomes over 60 times, with $PP^{\text{triadic}}_{\text{day 7}} = 1.7\text{K}$ and $PP^{\text{lda}}_{\text{day 7}} = 112\text{K}$. This observation exactly shows the volatility of items and the persistency of queries, thus motivating the idea of modeling ephemeral relations through a persistent layer. The purpose of comparing our approach with the standard LDA exposed to a sparse dyadic setting is not to claim a better statistical method, but rather to propose a general framework that can help realize the full potential of topic modeling.

7 Conclusion and Future Work

In this paper, we presented a method for building reliable topic models by using unstructured meta-data when direct topic modeling is infeasible due to the sparsity and volatility of co-occurrence patterns. We show both how to use the meta-data to densify the original input and to build a better topic model. Furthermore, we show how to efficiently estimate the model parameters in practice

by decoupling the interactions between the two original input sets. Most notably, this allows us to efficiently compute model parameters in the presence of new data without recomputing all model parameters. We demonstrate the usefulness of our method by deriving a statistical formulation of a novel personalized recommendation system for volatile items. Our data is challenging due to extreme sparsity of the input which prohibits the use of standard topic modeling and collaborative filtering techniques. We show that by using the search queries, we can still build a reliable topic model, which can then be used for efficient recommendation.

In the future, we would like to add a temporal aspect to our method to reflect the evolution of the latent topics and user preferences. This could include modeling short-term effects, such as topics that become prominent because of external events (e.g., Super Bowl), cyclic topic shifts that arise in certain seasons, as well as long term topic changes that appear, e.g., from new products introduced in the market. Moreover, we are planning to compare the efficiency and effectiveness of the various item recommendation methods. Finally, it is desired to benchmark our approach with standard LDA with a larger scale data set, e.g., three-month training data and one-month daily evaluation. We expect to further strengthen the advantages of our method.

Acknowledgments

The authors would like to thank Neel Sundaresan (eBay Inc.), as well as Yanen Li (University of Illinois at Urbana-Champaign) and Qing Xu (Vanderbilt University) for the help with acquiring and processing of the data, as well as for the many valuable comments and suggestions.

References

1. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Tr. on Knowl. and Data Eng. 17(6), 734–749 (2005)
2. Agarwal, D., Chen, B.: fLDA: Matrix factorization through latent Dirichlet allocation. In: Proc. WSDM (2010)
3. Bell, R., Koren, Y., Volinsky, C.: Modeling relationships at multiple scales to improve accuracy of large recommender systems. In: Proc. KDD, pp. 95–104 (2007)
4. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet allocation. J. Mach. Learn. Res. 3, 993–1022 (2003)
5. Canny, J.: GaP: a factor model for discrete data. In: Proc. SIGIR, pp. 122–129 (2004)
6. Chen, Y., Canny, J.: Probabilistic clustering of an item. U.S. Patent Application 12/694,885, filed with eBay (2010)
7. Chen, Y., Pavlov, D., Canny, J.: Large-scale behavioral targeting. In: Proc. KDD, pp. 209–218 (2009)
8. Das, A.S., Datar, M., Garg, A., Rajaram, S.: Google news personalization: scalable online collaborative filtering. In: Proc. WWW, pp. 271–280 (2007)

9. Fei-Fei, L., Perona, P.: A Bayesian hierarchical model for learning natural scene categories. In: CVPR, pp. 524–531 (2005)
10. Girolami, M., Kabán, A.: On an equivalence between pLSI and LDA. In: Proc. SIGIR, pp. 433–434 (2003)
11. Griffiths, T.L., Steyvers, M.: Finding scientific topics. Proc. Natl. Acad. Sci. U.S. 101, 5228–5235 (2004)
12. Hofmann, T.: Probabilistic latent semantic analysis. In: Proc. UAI, pp. 289–296 (1999)
13. Koren, Y.: Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: Proc. KDD, pp. 426–434 (2008)
14. Liu, Y., Zha, H., Qin, H.: Shape topics: A compact representation and new algorithms for 3d partial shape retrieval. In: Proc. CVPR, vol. 2, pp. 2025–2032 (2006)
15. Micarelli, A., Gasparetti, F., Sciarrone, F., Gauch, S.: Personalized search on the World Wide Web. In: The Adaptive Web, pp. 195–230 (2007)
16. Mimno, D., Wallach, H.M., Naradowsky, J., Smith, D.A., McCallum, A.: Polylingual topic models. In: Proc. EMNLP, pp. 880–889 (2009)
17. Steyvers, M., Griffiths, T.: Probabilistic topic models. In: Handbook of Latent Semantic Analysis, pp. 424–440 (2007)
18. Wetzker, R., Umbrath, W., Said, A.: A hybrid approach to item recommendation in folksonomies. In: Proc. WSDM, pp. 25–29. ACM, New York (2009)