A Topic Model-based Personalization over Time

El Mehdi Rochd*†, Mohamed Quafafou*
† Marketshot. Paris, France
*Aix-Marseille University. LSIS UMR 7296. Marseille, France
{el-mehdi.rochd;mohamed.quafafou}@univ-amu.fr

ABSTRACT

In the field of information retrieval, personalization is an opportunity that enables to exploit ranking algorithms with the aim of ensuring that the returned results correspond to the interests of searchers. Recently, probabilitstic topic models were successfully used to achieve the personalization task using query logs. Furthermore, it was established that the incorporation of user profiles in these systems improves significantly their performance due to prior knowledge of queries made by a user. However, the proposed models are static, while the data are often collected over time. Moreover, in the case of processing large volumes of data, that can extend over a wide period, it becomes difficult to classify all results that have appeared during this period. Then, personalization systems can not deal with new users. To over these limitations, we propose a model called the *Dynamic personalization Topic Model* (DpTM), that personalizes search dynamically and addresses the challenging problem of predicting results for new users. We compare our model, with recent topic models and used them to rank results by their likelihood given a particular user/query pair. Experiments on real data show the effectiveness of our approach.

Keywords

Topic Models, User profile, Personalization, E-commerce

1. INTRODUCTION

Personalization has been a very active area of research in recent years. Indeed, the construction of user profiles is an important component of any personalization system. Some common uses of personalization include customizing the appearance and content of many websites and creating search engines that exploit user profiles. Theses engines can play an important part in commercial applications.

However, search engines return results based primarily on the submitted queries. Nevertheless, the same query could be used in different contexts since individual users have different interests. To improve the relevance of search results, it is necessary to re-sort the ranked lists according to the learned user profile.

Thus, in order to enhance rankings, personalization adds to the conventional search systems, a component, which is the user profile, beyond merely his/her issued query. The key idea is that by understanding some information about the user issuing a query, we can adapt the ranked lists so that the likelihood of highly rated results being relevant is increased. Furthermore, many studies have subsequently shown that the greatest care must be taken when applying personalization to avoid damaging an already near-optimal ranked list [7, 20].

Recently, probabilitstic topic models [2, 19] were successfully used to achieve the personalization task using query logs, by making use of prior knowledge of the user's previous search behavior to infer his/her topical interests. However, in such models, the user's profile is considered static, when in fact, the user interests may change over time. Indeed, the majority of large data sets processed by topic models do not have static co-occurrence patterns. On the contrary, they are dynamic. The data collection is often over time, and in general, the patterns present in the first part of the collection are not in effect later. The prominence of topics is experiencing ups and downs. They can be separated or they can merge to form new topics. For example, in this paper, we use query logs from a real online products comparator which generates $25,000$ transactions per month, and we observe a net change in the proposed products every 3 months. Thus, it becomes necessary to capture the evolution in the user's interest, as this will enables a gain in precision and memory management since it is very difficult to pertinently classify the results when data cover very broad periods if the dynamic is not incorporated in the model.

In this paper, we propose to build a personalized and dynamic ranking model, where user profiles are constructed from the representation of the results that were selected by the users over a topic space. We use latent topic models to determine these topics by assuming that a topic is a probability distribution over the user's query (the query is composed of words). Therefore, the topic space is extracted directly from the query logs without requiring human intervention to define these topics as would be the case if we had used a human-generated ontology to specify the topics [18, 8]. We also propose to address a limit of personalization systems, which lies in the fact that they are not able to deal with new users. Our contribution consists on incorporating the dynamics of fluctuating interests over time into personalization with topic models, and dealing with new users.

We present experimental results with two real-world data sets based on real users. The first one is composed of queries from a website that compares online products based on users search criteria. The second one is the AOL Search dataset, which is a collection of real query log data. Our experiments performed on these data show that by introducing the user profiles and the dynamics in the ranking model, we can provide lists, the classification of which is greatly improved compared to the lists that can be generated by a static model.

The rest of the paper is organized as follows. In section 2, we present the related work. Section 3 describes our Dynamic personalization Topic Model. In Section 4, we discuss the experimental setting. The results and the model evaluation are discussed in Section 5. Section 6 brings this paper to the conclusion and future work.

2. RELATED WORK

The e-marketing techniques are used to explore the client relationships and the ongoing interaction. This interaction enables to discover the interests of users and consequently gives the information that may interest them at the right time. To achieve these ends, we should first gather information about users and build their profiles from the analysis of this information.

The approaches differ depending on the length of the used profile data and how the chosen data is turned into a suitable user profile. In [23], only the information from the current search session has been considered to build short-term profiles. In [17], the authors attempted to build longer-term user profiles. In [1], it was shown how these short and long-term profiles can be combined.

After the selection of prior interaction data, it must be converted into a user profile in order to form a representation of the user's interests. Different techniques can be used to generate these profiles. In [14], an approach which uses vectors of the original terms has been proposed. An other approach used in [16] aims to map the user's interests onto a set of topics, which can be defined by the users them-selves. Finally, an approach allows to extract these topics from large online ontologies of web sites, such as the Open Directory Project [6].

A new idea takes hold, consisting on using latent topic models to determine the topics instead of employing a human-generated ontology. In fact, topic models, such as latent Dirichlet allocation (LDA) [4], are hierarchical Bayesian models of discrete data. They have become an indisputable tool for exploratory and predictive analysis of text. They posit that a small number of distributions over words, called topics, can be used to explain the observed data.

In this context, new LDA-based models for the analysis of the problem of personalized search have been introduced in [5, 10]. The contribution of these works is the integration of a user/topic distribution in the graphical model, which means that the user plays a part in the generation process. The obtained results did not give the desired effect wherein personalization increases the performance. The authors have hypothesized that this negative effect on the ranking lists, might be due to the integration of the user in the generation process, which makes him/her very influential in the model and can be overwhelming information derived from the observed data, while it can be more useful. In [9], a model called the Personalized Topic Model (PTM) was presented for personalized search from query logs using sets of latent topics derived directly from the log files themselves, where the user is not included in the generation process, but rather is subtly introduced as part of the ranking formula. The authors have observed an improvement in performance compared to non-personalized models. In addition, this model is particularly effective in cases of sparse prior data where click frequencies can not be used to generate good ranked lists.

Concerning the topic models, most of them assume that the data are exchangeable in the collection, which means that their probability is time invariant. However, many data collections, evolve over time. In [3], the authors proposed the Dynamic Topic Model which uses a state space model on the natural parameters of the multinomial distributions that represent the topics. This model requires that the time is discretized into several periods, and within each period, LDA is used to model the data. To test this model, the authors have analyzed the journal Science from $1880-2002$, assuming that articles are exchangeable within each year.

In this paper, we present a model for dynamic personalized search, based on query logs using latent topics derived directly from data. In our model, the user is not included as part of the generative process but we have subtly introduced him/her within the ranking formula. Moreover, we propose an approach to deal with new users. Experiments conducted on real data demonstrate performance improvements compared to static models.

3. DYNAMIC PERSONALIZATION TOPIC MODEL

3.1 The model description

In this section, we present our Dynamic personalization Topic Model (DpTM), the graphical model of which is given in Figure 1. The model involves an observed resource $d \in \{1,...,D\}$ (product or URL), a latent topic variable $z \in \{1,...,Z\}$, an observed word $w \in \{1,...,W\}$ and an observed user $u \in \{1,...,U\}$. This structure is repeated for all words in a user's query, all queries made by the user and all users in the log file. The model is used to derive topic allocations

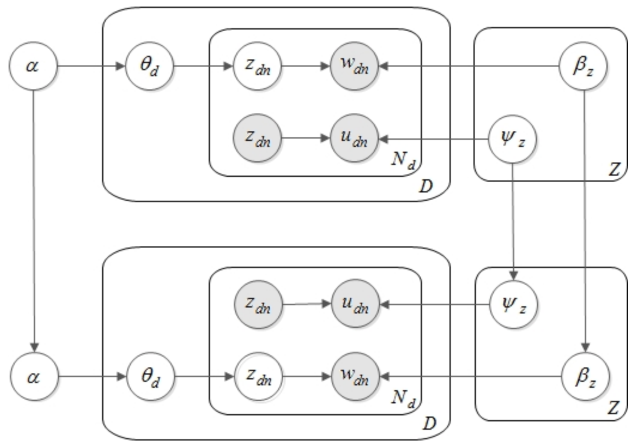


Figure 1: Graphical model of the Dynamic personalization Topic Model (for two time slices).

for each resource in the log file and to determine each user's topical interest profile.

The model parameters are: θ_d which is a probability vector over topics for each resource, β_z which is a probability vector over words (composing a query) for each topic, and $\psi_{u|z}$ which is a probability vector over users for each topic. Symmetric Dirichlet priors with hyperparameters α and γ are placed over θ and ψ in order to prevent them from overfitting the data. Regarding topics, when z is unshaded, then it is latent, and when it is shaded, then it has converged and thus becomes observable. Consequently, it can be used to identify the user's topical interests.

When the vertical arrows are removed, there is rupture of the dynamics, the graphical model reduces to a set of independent Personalization Topic Models. With time dynamics, the zth topic at slice t has evolved from the zth topic at slice $t-1$.

First, we recall the principle of the Dynamic Topic Model (DTM). In a time stamped data collection, the aim of this system is to model the observed changes in the latent topics through the course of the collection. For example, in the field of e-commerce, a single topic will evolve as the user's interests associated with it change. In the DTM, data are divided into sequential groups, and the topics of each slice evolve from the topics of the previous slice. We assume that data in a group are exchangeable and that a topic is represented as a distribution over the fixed vocabulary of the collection. This model assumes that the evolution of the natural parameters of the multinomial distributions that represent the topics is governed by a discrete-time state space model. This can be seen as a time-series extension to the logistic normal distribution [22].

However, the DTM is a non-personalized model. Thereby, we propose to extend it to consider both the dynamics and the topical interests of users. Thus, the generative process of the DpTM is represented by the following steps:

1. Draw topics $\beta_t|\beta_{t-1} \sim \mathcal{N}(\beta_{t-1}, \sigma^2 I)$
2. For each resource d, draw a multinomial θ_d from a Dirichlet prior α such that: $\alpha_t|\alpha_{t-1} \sim \mathcal{N}(\alpha_{t-1}, \delta^2 I)$
3. For each user u, draw a multinomial $\psi_{u|z}$ from a Dirichlet prior γ such that: $\gamma_t|\gamma_{t-1} \sim \mathcal{N}(\gamma_{t-1}, \rho^2 I)$
4. For each resource:
   - Draw $\eta_t \sim \mathcal{N}(\alpha_t, a^2 I)$
   - For each query element (word) w:
     - Draw $z \sim \text{Multinomial}(\Delta(\eta))$
     - Draw $w_{t,d,n} \sim \text{Multinomial}(\Delta(\beta_{t,z}))$
5. After the topics convergence, calculate the distribution $\psi^{(t)}_{u|z}$ over users for each topic at slice t in order to capture the user's interests at slice t taking into account the discovered dynamic topics.

The function Δ maps the multinomial parameters, which are unconstrained, to its mean parameters, which are on the simplex:

$$\Delta(\beta_{t,z})_w = \frac{\exp(\beta_{t,z,w})}{\sum_w \exp(\beta_{t,z,w})}$$

In the same manner, we consider a natural parameterization of the multinomial θ as follows:

$$\eta_i = \log \tfrac{\theta_i}{\theta_Z}$$

3.2 Approximate inference

As for the PTM, the true posterior of the DpTM is intractable. But, unlike the PTM case, we can not use the gibbs sampling because in the sequential setting, the distribution of words for each topic is not conjugate to the word probabilities. Thus, variational methods [11] are more appropriate for our model.

Variational methods aim to optimize the free parameters of a distribution over the latent variables q so that this distribution is close in Kullback-Liebler (KL) divergence to the true posterior p. Then, this distribution can be used as a substitute for the true posterior.

Since we do not include the user in the generative process, the variational distribution for a model with Z dynamic topics and T time windows, is the same as the distribution used in DTM:

$$\prod_{z=1}^{Z} q(\beta_{z,1}, ..., \beta_{z,T}|\widehat{\beta}_{z,1}, ..., \widehat{\beta}_{z,T})$$

x $\prod_{t=1}^{T} \left(\prod_{d=1}^{D_t} q(\theta_{t,d}|\mu_{t,d}) \prod_{n=1}^{N_{t,d}} q(z_{t,d,n}|\nu_{t,d,n})\right)$

where each proportion vector $\theta_{t,d}$ is endowed with a free Dirichlet parameter $\mu_{t,d}$ and each topic $z_{t,d,n}$ is endowed with a free multinomial parameter $\nu_{t,d,n}$ and $\widehat{\beta}$ variables are observations to a variational Kalman filter [13]. In fact, the resulting variational approximation for the natural topic parameters $\{\beta_{z,1}, ..., \beta_{z,T}\}$ incorporates the dynamics and the approximation can be done based on a Kalman filter. Therefore, our state space model is:

$$\beta_t|\beta_{t-1} \sim \mathcal{N}(\beta_{t-1}, \sigma^2 I)$$

$$w_{t,n}|\beta_t \sim \text{Mult}(\pi(\beta_t))$$

The variational state space model is formed as:

$$\widehat{\beta}_t|\beta_t \sim \mathcal{N}(\beta_t, \widehat{\epsilon}_t^2 I)$$

We next use the forward-backward algorithm for the Kalman filter to compute the expectations for updating the variational parameters.

As described in [3], the forward mean and variance of the variational posterior are:

$$m_t \equiv E[\beta_t|\widehat{\beta}_{1:t}]$$
$$= \left(\frac{\widehat{\epsilon}_t^2}{V_{t-1} + \sigma^2 + \widehat{\epsilon}_t^2}\right) m_{t-1} + \left(1 - \frac{\widehat{\epsilon}_t^2}{V_{t-1} + \sigma^2 + \widehat{\epsilon}_t^2}\right) \widehat{\beta}_t$$

$$V_t \equiv E[(\beta_t - m_t)^2|\widehat{\beta}_{1:t}]$$
$$= \left(\frac{\widehat{\epsilon}_t^2}{V_{t-1} + \sigma^2 + \widehat{\epsilon}_t^2}\right) (V_{t-1} + \sigma^2)$$

with initial conditions specified by fixed m_0 and V_0.

The backward recursion then calculates the marginal mean and variance of β_t given $\widehat{\beta}_{1:T}$ as:

$$\tilde{m}_{t-1} \equiv E[\beta_{t-1}|\widehat{\beta}_{1:T}]$$
$$= \left(\frac{\sigma^2}{V_{t-1} + \sigma^2}\right) m_{t-1} + \left(1 - \frac{\sigma^2}{V_{t-1} + \sigma^2}\right) \tilde{m}_t$$

$$\tilde{V}_{t-1} \equiv E[(\beta_{t-1} - \tilde{m}_{t-1})^2 | \widehat{\beta}_{1:T}]$$
$$= V_{t-1} + \left(\frac{V_{t-1}}{V_{t-1} + \sigma^2} \right)^2 \left(\tilde{V}_t - (V_{t-1} + \sigma^2) \right)$$

with initial conditions $\tilde{m}_T = m_T$ and $\tilde{V}_T = V_T$.

With this forward-backward computation in hand, we optimize the variational observations $\widehat{\beta}$. The minimization of the KL divergence is equivalent to the maximization of the bound on the likelihood of the observations using Jensen's inequality [11]:

$$\mathcal{L}(\widehat{\beta}) \geq \sum_{t=1}^{T} E_q[\log p(\boldsymbol{w}_t | \beta_t)] + E_q[\log p(\beta_t | \beta_{t-1})] + H(q) \tag{1}$$

where $H(q)$ is the entropy.
Finally, to optimize the variational observations, we need to compute the derivative $\dfrac{\partial \mathcal{L}}{\partial \widehat{\beta}_{t,w}}$.

For more details on the calculations, see [3].
This optimization gives us the dynamic topics. Once they converge, we use them as well as the user's profile to capture his/her dynamic topical interests. We consider then, that the user does not play a part in the generation process of the model. This is why we have placed in the graphical model, a distribution which we call the user/topic distribution, that will be used after obtaining the dynamic topics.

3.3 Calculation of the user/topic distribution

Since we are in the case where the variables are observed (topics which have converged and users), we use the maximum likelihood method to estimate $\psi_{u|z}$ (the user/topic distribution). Indeed, in the case of Bayesian estimation, the objective is to find the most likely parameters ψ given the observed data using a priori parameters.

Bayes rule gives us:

$$L = p(\psi | u, z) \propto p(u, z | \psi) p(\psi)$$
$$\propto p(u | z, \psi) p(\psi)$$

Since ψ is a multinomial distribution, its conjugate prior distribution is a Dirichlet distribution whose coefficient is γ. Thus, for Z topics and U users, L becomes:

$$L = \prod_{z=1}^{Z} \prod_{u=1}^{U} \psi_{u|z}^{N_{uz}} \prod_{z=1}^{Z} \prod_{u=1}^{U} \frac{\Gamma(U\gamma)}{\Gamma(\gamma)^U} \psi_{u|z}^{\gamma_z - 1}$$
$$= \frac{\Gamma(U\gamma)}{\Gamma(\gamma)^U} \prod_{z=1}^{Z} \prod_{u=1}^{U} \psi_{u|z}^{N_{uz} + \gamma_z - 1}$$

where Γ is the gamma function and N_{uz} is the count denoting the number of times the topic z appears together with the user u.
Taking the logarithm of L, we obtain:

$$\log L = \log \frac{\Gamma(U\gamma)}{\Gamma(\gamma)^U} + \sum_{z=1}^{Z} \sum_{u=1}^{U} (N_{uz} + \gamma_z - 1) \log \psi_{u|z} \tag{2}$$

To simplify the calculations, we assume that the Dirichlet coefficients are equal:

$$\gamma_1 = \gamma_2 = ... = \gamma_Z = \gamma$$

We know that: $\sum_{u=1}^{U} \psi_{u|z} = 1$.
Thereby:

$$\psi_{U|z} = 1 - \sum_{u=1}^{U-1} \psi_{u|z}$$

By injecting the last two equations in equation (2), we get:

$$\log L = \log \frac{\Gamma(U\gamma)}{\Gamma(\gamma)^U} + \sum_{z=1}^{Z} \Big(\sum_{u=1}^{U-1} (N_{uz} + \gamma - 1) \log \psi_{u|z}$$
$$+ (N_{Uz} + \gamma - 1) \log(1 - \sum_{u=1}^{U-1} \psi_{u|z}) \Big)$$

By taking the derivative of this term with respect to $\psi_{u|z}$, we obtain:

$$\frac{\partial \log L}{\partial \psi_{u|z}} = \frac{N_{uz} + \gamma - 1}{\psi_{u|z}} - \frac{N_{Uz} + \gamma - 1}{1 - \sum_{u=1}^{U-1} \psi_{u|z}}$$
$$= \frac{N_{uz} + \gamma - 1}{\psi_{u|z}} - \frac{N_{Uz} + \gamma - 1}{\psi_{U|z}}$$

We set this term to zero in order to get the maximum of $\psi_{u|z}$ that we denote $\widehat{\psi}_{u|z}$, we obtain:

$$\frac{N_{1z} + \gamma - 1}{\widehat{\psi}_{1|z}} = \frac{N_{2z} + \gamma - 1}{\widehat{\psi}_{2|z}} = ... = \frac{N_{Uz} + \gamma - 1}{\widehat{\psi}_{U|z}}$$
$$= \frac{\sum_{u=1}^{U} (N_{uz} + \gamma - 1)}{\sum_{u=1}^{U} \widehat{\psi}_{u|z}} = \sum_{u=1}^{U} (N_{uz} + \gamma - 1)$$

Thus:

$$\frac{N_{uz} + \gamma - 1}{\widehat{\psi}_{u|z}} = \sum_{u=1}^{U} (N_{uz} + \gamma - 1)$$

Finally, we get the expression of the user/topic distribution:

$$\widehat{\psi}_{u|z} = \frac{N_{uz} + \gamma - 1}{\sum_{u=1}^{U} (N_{uz} + \gamma - 1)} \tag{3}$$

This equation will be used next to rank resources according to the user's query.

3.4 Predicting resources for new users

The limit of personalization systems is their inability to handle queries of new users. We propose the following approach to overcome this limitation:

1. For each new user, generate his/her distribution over the query elements (vocabulary containing words composing all users'queries) using LDA.
2. Calculate the probability distribution of former users over the query elements (the same vocabulary size).
3. Calculate the KL divergence between a new user's distribution over query elements and each distribution of former users over the same vocabulary.
4. Select the former user u^{former} for which the KL divergence is the lowest.

5. Predict resources for the new user using his/her query and the user/topic distribution of the selected u^{former}.

3.5 Ranking Online Resources

In this section, we describe formulas for ranking online resources (products and/or URLs) using the parameters that were estimated based on the DpTM and the other models. The ojective is to return to the user a ranked set of resources ($d \in \mathcal{D}$) according to their likelihood given the user's query $q = \{w_1, w_2, ..., w_n\}$ under each model. Initially, we recall the formula in the case of a non-personalized model (LDA):

$$\begin{aligned} p(d|q) \propto p(d)p(q|d) &= p(d) \prod_{w \in q} p(w|d) \\ &= \widehat{\pi}_d \prod_{w \in q} \sum_z p(w|z)p(z|d) \end{aligned} \tag{4}$$

where: $\widehat{\pi}_d = p(d) = \frac{N_d}{N}$, N_d is the number of words composing the user's query, which led to the selection of resource d and N is the total number of words composing all user queries.

The ranking formula consists of multiplying a prior on the probability of the resource (which we denote $p(d)$) with the probability of the query given the resource (which we denote $p(q|d)$). This latter quantity can be estimated by introducing latent topics from topic models. Indeed, topic models allow to estimate the probability of words given topics $p(w|z)$ and the probability of topics given resources $p(z|d)$.

In the case of the PTM, we know the queries issued by a user. Thus, the user's preferences can be included into the ranking formula. This means that we rank resources according to their likelihood given both the query and the user who made this query as follows:

$$\begin{aligned} p(d|q,u) &\propto p(d) \prod_{w \in q} p(w,u|d) \\ &= p(d) \prod_{w \in q} \sum_z p(w|z)p(u|z)p(z|d) \end{aligned}$$

This personalization model was extended by introducing an additional parameter λ in the range zero to one, which was used to weight the probability of a user given a particular topic $p(u|z)$ as follows:

$$\tilde{p}(d|q,u) = \widehat{\pi}_d \prod_{w \in q} \sum_z p(w|z)p(u|z)^{\lambda} p(z|d)$$

The introduction of this new parameter is motivated by the fact of being able to control the amount of influence that the user's topical interests may have on the ranking.

Concerning the DpTM, we also know the time window at which the user has made his/her query, which we include into the ranking formula, in addition to that user's prefences. Thus, we rank resources according to their likelihood given the query, the user and the time window at which the user has made his/her query as follows:

$$\begin{aligned} p^{(t)}(d|q,u) &\propto p^{(t)}(d) \prod_{w \in q} p^{(t)}(w,u|d) \\ &= p^{(t)}(d) \prod_{w \in q} \sum_z p^{(t)}(w|z)p^{(t)}(u|z)p^{(t)}(z|d) \end{aligned}$$

Thus, the resources are ranked according to:

$$\text{score}^{(t)}(d,q,u) = \widehat{\pi}_d^{(t)} \prod_{w \in q} \sum_z \widehat{\beta}_{w|z}^{(t)} \widehat{\psi}_{u|z}^{\lambda(t)} \widehat{\theta}_{z|d}^{(t)} \tag{5}$$

4. EXPERIMENTS

In this section, we describe the experiments we have conducted on two real-world datasets for personalizing search.

4.1 Datasets

4.1.1 Marketshot dataset

Marketshot[1] is a company that generates qualified leads on the Internet through its websites. The dataset used in this paper is from the query logs of one of its website that connects potential buyers with major brands and distribution networks in the market of mobile telephony[2]. This website provides information about available products in the market of mobile telephony. It also provides various search and comparison tools in order to help the undecided users to choose the package that interests them the most among the multitude of available products. The data are based on a 4-months web log file from December, 2013 to March, 2014. We generate the training data automatically from log file without any human intervention. To clean the data, we have kept the queries which resulted in a product selection. Then, we have selected only the products for which more than 10 users had clicked on at least once. Moreover, we selected only the users with more than 10 remaining queries. This preprocessing stage is carried out to ensure that users have made a significant number of queries and that products were also selected reasonably. The resulting reduced dataset is described in more detail in Table 1. Our log file is composed mainly of 7 attributes: the ID of the transaction, the ID of the user session, the mobile provider, the package, the package features, the user's query and the date when the user has made his/her query. Table 2 shows an example of 3 transactions from this query log. In our experiments, we consider that a product is represented by the triplet: (Package, Mobile Provider, Package features).

4.1.2 AOL dataset

This publicly available query log dataset [15] is provided to the research community by AOL search engine[3]. It consists of ten files containing nearly 37M lines of data representing 657k users. We focus on the search events happened from March to May, 2006. Each search event is represented by a tuple (u, q, t), which means user u issued query q at time t, and we sort all the search events by their time. Then, we normalize queries through punctiation-removal and casefolding. User privacy was protected by analysing results only over aggregate data. The same preprocessing stage as above was performed for this dataset. In fact, since this dataset is huge, we considered arbitrarily one out of the ten available files and we selected the queries which resulted in a click on a URL. Then, the URLs for which more than 100 users had clicked on at least once were selected. Finally, we selected only those users with more than 100 remaining queries. The resulting reduced dataset is described in more detail in Table 1.

[1]http://www.marketshot.fr
[2]http://www.choisirsonforfait.com/
[3]http://search.aol.com

Marketshot Dataset		AOL Dataset	
Queries	1,933	Queries	65,616
Users	130	Users	1,190
Products	174	URLs	237
Vocabulary size	223	Vocabulary size	22,162

Table 1: Datasets features.

Id	Id session	Package	Mobile Provider	Package features	date	user's request
3	73f08ee8	Mobile plan 1	Mobile Provider A	2-years contract	2013-01-15 11:57:22	1 hour of calls, cell phone
2	ce77d6fb	Mobile plan 2	Mobile Provider B	2 hours plan	2013-01-15 11:57:15	30 minutes of calls
1	08f43fc9	Mobile plan 3	Mobile Provider C	unlimited calls	2013-01-15 11:56:43	unlimited calls and sms

Table 2: Log file format.

4.2 Methodology

The cleaned data is separated in two subsets: training subset ($\sim$ 95% of data) and testing subset ($\sim$ 5% of data). We have selected the last queries of each user for testing, to respect the order in which the queries were made. This means that the training and testing subsets follow the same chronological order.

Concerning the parameter setting, we set the hyperparameters α and γ to be $50/Z$ and 0.1 respectively, where Z is the number of topics.

We calculate the scores defined above for a user/query pair and for a specific time window in the cases of DpTM and DTM, and rank the resources accoding to the values of these scores. We consider that a ranked resource is relevant if it is the same one the user had actually selected. By doing this, we introduce the user profile in the analysis of the results, since the ranking relevance depends on the user and it is not generated by evaluators as it is the case for other approaches.

We evaluate the rankings by calculating three standard measures in the field of information retrieval: the Precision, the Mean Reciprocal Rank and the Mean Average Precision. We report these measures up to rank 10, since in information retrieval, it is valuable that pertinent resources appear early in the ranked lists.

In order to make the results more significant, we propose to evaluate 8 models:

- LDA as a static non-personalized model.
- PTM as a static personalized model.
- DTM-d as a dynamic non-personalized model with a 1-day time window.
- DTM-w as a dynamic non-personalized model with a 1-week time window.
- DTM-m as a dynamic non-personalized model with a 1-month time window.
- DpTM-d as a dynamic personalized model with a 1-day time window.
- DpTM-w as a dynamic personalized model with a 1-week time window.
- DpTM-m as a dynamic personalized model with a 1-month time window.

On the other hand, to evaluate the ranking relevance globally rather than the relevance of the top-10 results, we use the Canberra distance [12]. In addition, we assess the impact of the parameter λ introduced to allow control over the amount of influence the user profile has on the results'scores, as well as the impact of the difficulty of queries.

Finally, in order to determine if the dynamics is improving the ranking performance, we report a metric that we call the dynamic personalization gain (which we denote dP-Gain). This metric compares the number of times the DpTM improves the ranking (which we denote $\#better$) to the number of times it worsens it (which we denote $\#worse$). A simple expression of this equation is given by:

$$\text{dP-Gain} = \frac{\#better - \#worse}{\#better + \#worse}$$

When the value of this metric is 0, then there is no change between the DpTM and the other models, when it is positive, this means that our model improves the ranking and when it is negative, the ranking is deteriorated.

5. RESULTS

5.1 Top-K products-based evaluation

Tables 3 and 4 show the results of the ranking experiments for the 8 models on the two datasets. Firstly, we notice a net improvement in the performance of DpTM compared to the PTM, especially when considering a 1-day time window. This improvement is shown through the three measures introduced above. This result can be explained by the fact that our model is Markovian, ie the topics at time t have evolved and enriched from the topics at time $t-1$. Moreover, the granularity has an important role in improving the performance. In fact, the model quality is better for restricted time windows since this allows to regularly update topics and consequently the user's interests.

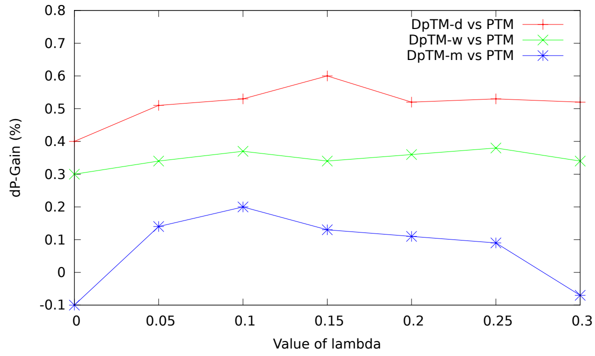
Secondly, we see a clear improvement in the mean reciprocal rank. We recall that the reciprocal rank of a query is the multiplicative inverse of the rank of the first correct answer and that the mean reciprocal rank is the average of the reciprocal ranks of results for a set of queries. This means that if the first relevant product is first-ranked, then the reciprocal rank is equal to 100%, and if the first relevant product is second-ranked, then the reciprocal rank is equal to 50%. The mean reciprocal rank obtained by the DpTM exceeds 70%, which means that the resource it proposes to the user is broadly either ranked first or second. In addition, to avoid

Measures	Models	Number of Topics									
		10	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**
Precision (%)	**LDA**	31.38	32.37	35.95	36.82	37.63	36.83	37.51	39.87	36.72	36.38
	DTM-d	72.89	73.72	73.20	72.99	73.22	73.07	72.06	71.20	**72.84**	72.81
	DTM-w	38.20	39.41	38.10	37.78	41.55	39.20	39.96	40.86	38.85	37.32
	DTM-m	35.20	38.71	36.50	36.89	35.33	37.78	37.68	39.06	37.55	36.37
	PTM	30.81	33.58	37.81	39.67	38.23	36.67	39.88	38.89	41.51	37.72
	DpTM-d	**75.54**	**74.84**	**73.31**	**74.29**	**73.39**	**73.49**	**74.10**	**72.25**	72.31	**73.61**
	DpTM-w	40.11	40.36	39.86	39.38	43.69	39.80	40.28	38.55	42.85	41.13
	DpTM-m	35.39	36.16	37.99	37.55	39.15	38.99	38.11	38.58	37.77	36.21
Mean Reciprocal Rank (%)	**LDA**	43.04	55.95	55.80	62.64	63.69	65.67	58.13	67.22	63.51	55.71
	DTM-d	73.81	76.27	76.13	**77.29**	76.50	76.45	75.31	70.95	72.91	74.85
	DTM-w	51.77	55.11	60.32	67.38	61.44	65.26	69.36	68.78	67.62	64.53
	DTM-m	49.67	53.92	59.95	61.97	58.15	64.30	65.16	65.47	63.60	59.87
	PTM	45.46	56.32	65.98	68.46	65.17	66.49	59.69	67.66	70.27	61.19
	DpTM-d	**78.57**	**79.22**	**77.05**	77.20	**77.32**	**77.81**	**75.78**	**77.60**	**73.61**	**76.72**
	DpTM-w	56.77	57.80	67.58	70.40	70.30	65.09	69.94	70.51	71.24	67.55
	DpTM-m	52.19	53.77	62.35	67.55	68.40	64.09	65.98	68.22	70.58	61.06
Mean Average Precision (%)	**LDA**	38.36	45.07	47.25	47.12	54.49	51.12	48.43	54.63	49.83	51.14
	DTM-d	75.04	77.71	77.66	77.02	76.46	77.51	76.42	73.25	74.88	75.63
	DTM-w	42.41	46.69	47.94	49.75	60.02	62.47	55.28	56.66	55.28	54.03
	DTM-m	39.88	45.75	45.33	48.51	54.72	52.37	53.17	54.85	49.47	48.96
	PTM	39.49	47.70	51.54	47.68	55.66	53.97	51.14	52.99	57.10	53.48
	DpTM-d	**78.88**	**78.65**	**78.44**	**78.20**	**77.88**	**78.47**	**76.53**	**78.03**	**75.12**	**77.42**
	DpTM-w	45.76	48.09	53.96	55.92	63.91	65.04	59.23	56.88	57.88	57.08
	DpTM-m	40.83	46.95	50.29	50.55	58.33	57.59	55.38	54.97	55.72	51.22

Table 3: Non-Personalized Models (LDA, DTM-d, DTM-w, DTM-m) vs Personalized Models (PTM, DpTM-d, DpTM-w, DpTM-m): Ranking performance of the 8 models on the test set over all Marketshot queries ($\lambda = 0.15$).

Measures	Models	Number of Topics									
		10	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**
Precision (%)	**LDA**	44.16	41.44	42.84	46.17	45.69	45.14	44.73	45.80	45.11	44.81
	DTM-d	64.28	66.57	65.83	68.81	69.14	68.15	67.77	66.39	65.18	65.27
	DTM-w	53.22	55.40	55.18	56.72	55.14	55.85	56.90	54.48	52.26	52.45
	DTM-m	45.84	44.74	45.72	47.28	46.24	45.38	44.64	44.82	43.65	44.27
	PTM	48.19	47.50	51.92	55.56	55.21	55.38	53.18	52.84	51.19	50.77
	DpTM-d	**72.44**	**72.18**	**72.51**	**73.20**	**72.56**	**72.11**	**71.74**	**71.92**	**70.85**	**70.17**
	DpTM-w	56.72	56.10	57.28	58.48	58.72	57.14	57.28	56.75	56.27	56.88
	DpTM-m	49.57	50.10	50.94	51.24	51.86	51.18	50.75	50.16	49.83	48.17
Mean Reciprocal Rank (%)	**LDA**	38.04	41.25	42.18	44.84	44.17	42.57	42.29	41.63	40.94	40.55
	DTM-d	61.27	61.98	62.45	63.20	62.86	62.42	61.75	61.49	60.96	60.63
	DTM-w	50.82	51.42	51.38	52.76	54.29	55.61	54.74	53.62	53.54	52.18
	DTM-m	46.80	46.17	46.88	48.10	49.35	49.20	48.38	47.73	46.81	45.94
	PTM	39.78	41.27	42.98	44.75	45.68	46.15	46.74	45.86	45.26	45.58
	DpTM-d	**64.85**	**65.36**	**66.01**	**67.19**	**67.92**	**67.26**	**66.29**	**66.18**	**65.75**	**65.34**
	DpTM-w	55.26	56.37	56.28	57.48	57.91	57.65	56.43	56.03	55.75	54.83
	DpTM-m	51.54	51.79	52.36	52.15	53.47	54.08	54.74	54.18	53.56	53.28
Mean Average Precision (%)	**LDA**	45.73	46.19	48.38	49.82	51.31	51.08	50.25	51.26	50.92	50.48
	DTM-d	66.18	66.73	67.58	69.01	69.72	69.21	68.58	68.11	67.80	67.62
	DTM-w	55.89	56.17	57.28	58.98	60.21	61.50	60.74	59.84	58.72	58.65
	DTM-m	49.11	50.73	51.28	52.37	52.04	51.82	51.10	50.86	49.68	49.12
	PTM	50.12	51.64	52.33	53.61	54.73	53.81	53.10	52.76	52.09	52.18
	DpTM-d	**73.10**	**74.05**	**74.38**	**74.79**	**73.41**	**73.05**	**72.40**	**72.15**	**71.58**	**71.26**
	DpTM-w	54.94	55.48	55.20	56.77	57.24	57.11	56.84	56.22	55.76	55.42
	DpTM-m	50.10	50.56	51.87	50.91	51.28	51.64	51.30	50.96	50.12	50.82

Table 4: Non-Personalized Models (LDA, DTM-d, DTM-w, DTM-m) vs Personalized Models (PTM, DpTM-d, DpTM-w, DpTM-m): Ranking performance of the 8 models on the test set over all AOL queries ($\lambda = 0.15$).



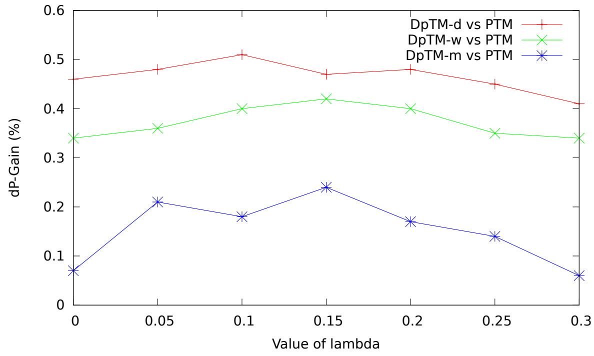


Figure 2: The effect of varying the λ parameter on the ranking algorithms: (Left) Results for the Marketshot dataset (Right) Results for the AOL dataset.

varying the number of topics in each experiment, we propose to use the Hierarchical Dirichlet Processes (HDP) [21] in order to automatically infer the number of topics. This enabled us to obtain 20 topics with respect to the Marketshot dataset and 40 topics with respect to AOL dataset. For the following experiments, we will consider these numbers of topics when running the models.

5.2 Global ranking evaluation

In order to analyze the relevance of the ranked resources globally and not only the top-10 products, we propose to evaluate the ranked products/URLs obtained from the PTM and the DpTM using the Canberra distance. In fact, the Canberra distance is a metric that measures the disarray for ranking lists, where rank differences in the top of the lists are more penalized than those at the end of the lists. Given two real-valued vectors $l, m \in \mathbb{R}^n$, their Canberra distance is defined as follows:

$$\text{Ca}(l,m) = \sum_{i=1}^{N} \frac{|l_i - m_i|}{|l_i| + |m_i|} \quad (6)$$

First, we compute the Canberra distance between the ranking given by the PTM and the ground truth. Second, we compute the Canberra distance between the ranking given by the DpTM and the ground truth (by varying the time window). The ground truth is obtained by considering resources that had been actually selected by users. Then, we calculate the difference between the two Canberra distances (PTM vs DpTM-d; PTM vs DpTM-w and PTM vs DpTM-m) in order to capture the number of times the DpTM is closer to the ground truth than the PTM. Table 5 shows the obtained results taking into account 3 time windows.

Datasets	DpTM-d	DpTM-w	DpTM-m
Marketshot	39.78 %	29.17 %	13.04 %
AOL	34.06 %	30.49 %	22.32 %

Table 5: Gain evolution according to the Canberra distance.

The results show that the global ranking has improved whatever the time window. However, we notice again a clear improvement when considering a 1-day time window.

5.3 The effect of λ

The parameter λ was introduced into the ranking formula for the personalized model to allow control over the amount of influence the user profile has on the products scores. We tested the effect of this parameter within the range of $\{0, 0.05, ..., 0.30\}$, where the extreme setting $\lambda = 0$ should collapse the DpTM back to the same estimates as DTM, and the PTM will have the same estimates as LDA. In this experiment, we make use of models consisting of 20 topics for the Marketshot dataset and 40 topics for the AOL dataset. Figure 2 shows the obtained result. We notice that, for almost all values of λ, the dynamic approach gives better performance for every time window, particularly when $\lambda = 0.15$. Indeed, our model can achieve a performance gain of 60%.

5.4 The effect of the query difficulty

In this section, we analyze the influence of the query difficulty on the performance improvement. When a given user/query pair had been observed before, we can use this information about prior clicks by assuming that the user will again click on the same results as before. However, in almost cases, the user/query pair will be novel and we will not have such prior information to exploit. When the query has been observed many times before, but always by other users, we are still able to use this information to provide a good ranking. We introduce a measure called the *click entropy* to identify such unambiguous queries. The click entropy of an observed query q is defined as follows:

$$H_q = \sum_{d \in D(q)} -p(d|q) \log_2 p(d|q)$$

where $D(q)$ is the set of the selected results given the query q and $p(d|q)$ is the frequency with which resource d was clicked amongst all the clicked resources given the query q. The entropy values vary in the range zero to the logarithm of the number of distinct resources clicked on for a query. Consequently, the range of values depends on the query. This makes the comparison of click entropy values accross queries complicated. To deal with this issue, we will use, in our experiments, normalized entropy values instead, where the range of values is limited to $[0, 1]$. This new measure is defined as follows:

$$\widehat{H}_q = \frac{H_q}{\log_2 |D(q)|}$$

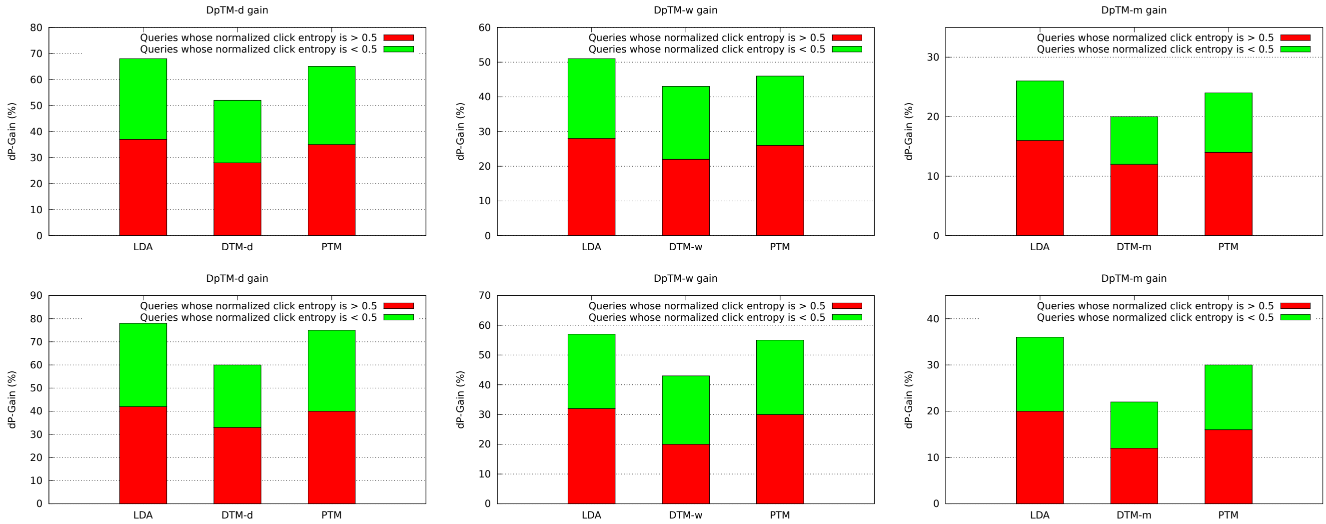


Figure 3: The effect of query ambiguity on the ranking algorithms taking into account the time window: (Top) Results for the Marketshot dataset (Bottom) Results for the AOL dataset.

We calculated this measure for all queries. We separated the queries into two groups: queries for which this measure is lower than 0.5 and queries for which this measure is greater than 0.5. Then we calculated the dP-Gain for each of the two groups that contain test queries.
Figure 3 shows how the performance of the DpTM changes as the normalized click entropy of the queries evolves.

We notice that the dP-Gain increases as the click entropy increases. In fact, the dP-Gain for the DpTM-d reaches 40% for queries whose normalized click entropy is greater than 0.5 and it drops to 15% for queries whose normalized click entropy is lower than 0.5.

We note the same observations concerning the other time windows, but with lower values of the gain. The conclusion of this experimentation is that the dP-Gain increases as the normalized click entropy increases and that the time dyanmics enables performance improvement.

5.5 Predictions for new users

Unlike the first experiment where the personalization task required a particular separation of data (users in the test set must have appeared in the training set), in this section, we divide the data randomly ($\sim$ 95% for training, $\sim$ 5% for testing). Then, we apply the procedure described in section 3.4 and we compare the DpTM-d to LDA and DTM-d since the best performances are obtained when using a 1-day time window model (PTM can not perform this task). We aplied again the HDP to automatically determine the number of topics, we found 18 topics for the Marketshot dataset and 34 topics for the AOL dataset. Again, we compute the precision, the mean reciprocal rank and the mean average precision. We report these measures up to rank 10. Table 6 shows the obtained results. We notice that our model can not only overcome the PTM limitation, but can also predict resources for users who have not previously issued any query.

6. CONCLUSION AND FUTURE WORK

In this paper, we have proposed a dynamic topic model that builds user profiles according to their search criteria on two real-world websites. These profiles are then used to personalize search.

The contribution of this work is on the one hand to model both the user's interests and the dynamics, and on the other hand, to deal with new users.

The experiment was conducted on two different datasets to treat search queries which extend over wide areas of research, with a rich array of different topics in which users are likely to be searching over an extended period of time with changing interests.

We observed that the user's interests change over time and the proposed model gives a great advantage in terms of performance, memory management and news updating.

We compared the DpTM with other models and evaluated its performance in terms of efficiency ranking. The results show that our model outperforms the others, especially when considering a 1-day time window, due to the Markovian nature of the DpTM.

We also evaluated performances of our model using a query difficulty metric, which is the click entropy. Again, our model performs well with a gain of up to 40% for queries for which the normalized click entropy is greater that 0.5.

Finally, the λ parameter which was introduced to control the amount of influence that the user's topical interests have on the ranking, enables to get a gain of up to 60%.

In our future work, we plan to analyze other families of function, which allow to control the influence of the user's topical interests on the ranking, with the objective to improve the gain. In addition, we intend to introduce time dynamics under a non-Markovian fashion.

7. REFERENCES

[1] P. N. Bennett, R. W. White, W. Chu, S. T. Dumais, P. Bailey, F. Borisyuk, and X. Cui. Modeling the impact of short- and long-term behavior on search personalization. In *Proceedings of the 35th International Conference on Research and Development in Information Retrieval*, SIGIR, pages

Measures	Models	Marketshot Dataset	AOL Dataset
Precision (%)	**LDA**	35.78	32.79
	DTM-d	72.68	64.35
	DpTM-d	**74.64**	**66.75**
Mean Reciprocal Rank (%)	**LDA**	39.61	33.56
	DTM-d	73.41	69.35
	DpTM-d	**76.31**	**71.16**
Mean Average Precision (%)	**LDA**	32.24	31.89
	DTM-d	74.58	70.53
	DpTM-d	**77.11**	**72.02**

Table 6: KL-based postprocessing: Ranking performance of the 3 models on the test sets over all queries ($\lambda = 0.15$)

185–194, 2012.

[2] D. M. Blei. Probabilistic topic models. *Commun. ACM*, 55(4):77–84, Apr. 2012.

[3] D. M. Blei and J. D. Lafferty. Dynamic topic models. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML, pages 113–120, 2006.

[4] D. M. Blei, A. Y. Ng, M. I. Jordan, and J. Lafferty. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[5] M. J. Carman, F. Crestani, M. Harvey, and M. Baillie. Towards query log based personalization using topic models. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM, pages 1849–1852, 2010.

[6] P. A. Chirita, W. Nejdl, R. Paiu, and C. Kohlschütter. Using odp metadata to personalize search. In *Proceedings of the 28th Annual International Conference on Research and Development in Information Retrieval*, SIGIR, pages 178–185, 2005.

[7] Z. Dou, R. Song, and J.-R. Wen. A large-scale evaluation and analysis of personalized search strategies. In *Proceedings of the 16th International Conference on World Wide Web*, WWW, pages 581–590, 2007.

[8] S. Gauch, J. Chaffee, and A. Pretschner. Ontology-based personalized search and browsing. *Web Intelligence and Agent Systems*, 1(3-4):219–234, Dec. 2003.

[9] M. Harvey, F. Crestani, and M. J. Carman. Building user profiles from topic models for personalised search. In *Proceedings of the 22nd ACM International Conference on Information and Knowledge Management*, CIKM, pages 2309–2314, 2013.

[10] M. Harvey, I. Ruthven, and M. J. Carman. Improving social bookmark search using personalised latent variable language models. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, WSDM, pages 485–494, 2011.

[11] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233, Nov. 1999.

[12] G. Jurman, S. Riccadonna, R. Visintainer, and C. Furlanello. Canberra distance on ranked lists. In S. Agrawal, C. Burges, and K. Crammer, editors, *Proceedings of Advances in Ranking NIPS 09 Workshop*, pages 22–27, 2009.

[13] R. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[14] N. Matthijs and F. Radlinski. Personalizing web search using long term browsing history. In *Proceedings of the Fourth International Conference on Web Search and Data Mining*, WSDM, pages 25–34, 2011.

[15] G. Pass, A. Chowdhury, and C. Torgeson. A picture of search. In *Proceedings of the 1st International Conference on Scalable Information Systems*, Infoscale, 2006.

[16] A. Pretschner and S. Gauch. Ontology based personalized search. In *Proceeding of the International Conference on Tools with Artificial Intelligence*, ICTAI, pages 391–398, 1999.

[17] F. Qiu and J. Cho. Automatic identification of user interest for personalized search. In *Proceedings of the 15th International Conference on World Wide Web*, WWW, pages 727–736, 2006.

[18] A. Sieg, B. Mobasher, and R. Burke. Web search personalization with ontological user profiles. In *Proceedings of the 16th Conference on Conference on Information and Knowledge Management*, CIKM, pages 525–534, 2007.

[19] M. Steyvers and T. Griffiths. *Probabilistic Topic Models*. In T. Landauer, D. Mcnamara, S. Dennis, W. Kintsch. Latent Semantic Analysis: A Road to Meaning. Laurence Erlbaum, 2007.

[20] J. Teevan, S. T. Dumais, and E. Horvitz. Potential for personalization. *ACM Trans. Comput.-Hum. Interact.*, 17(1):4:1–4:31, Apr. 2010.

[21] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101, 2004.

[22] M. West and J. Harrison. *Bayesian Forecasting and Dynamic Models (2nd Ed.)*. Springer-Verlag New York, Inc., 1997.

[23] R. W. White, P. Bailey, and L. Chen. Predicting user interests from contextual information. In *Proceedings of the 32nd International Conference on Research and Development in Information Retrieval*, SIGIR, pages 363–370, 2009.