A nonparametric mixture model for personalizing web search

El Mehdi Rochd[1,2] and Mohamed Quafafou[1]

[1] Aix-Marseille University, LSIS UMR 7296, France
[2] Marketshot, Paris, France
{el-mehdi.rochd,mohamed.quafafou}@univ-amu.fr

Abstract. Probabilistic topic models were successfully used to achieve the personalization task using query logs. Thus, both users and previously clicked results are considered when estimating probability distrubutions in order to answer users'queries. However, the proposed models are generally parametric and require to define in advance the number of topics. Moreover, they can not deal with new users. To overcome these limitations, we propose a model called the Hierarchical personalized Dirichlet Processes (HpDP) that personalizes search and allows to automatically learn the number of latent topics. It also addresses the challenging problem of predicting results for new users. We compare our model, with recent topic models and use them to rank online products by their likelihood given a particular user/query pair. Experiments performed on data from a real online products comparator show the effectiveness of our approach.

1 Introduction

Building user profiles is an important component of personalization systems. In fact, in commercial applications, personalization relies on user profiles to help adapting the content of websites in order to propose information that best fits the user's interests. To achieve these ends, we should first gather information about users and build their profiles from the analysis of this information. Many reported approaches enable to get necessary knowledge about the users in order to build their profiles. An approach consists of considering information from the current search session to build short term profiles [10]. In [8], an approach attempted to build long-term user profiles. In [1], the authors have shown how these short and long-term profiles can be combined. Once prior interaction data are selected, the following step is to convert it into a user profile in order to perform a representation of the user's interests. Different techniques enable to generate these profiles. The authors of [6] adapted an approach using vectors of the original terms. Another approach, described in [7], aims to map the user's interests onto a set of topics, which can be defined by the users themselves. Then, an additional approach enables to extract these topics from large online ontologies of websites, such as the Open Directory Project [3].

A new technique that starts to arouse interest, consists in using latent topic models [9] to determine these topics instead of employing a human-generated ontology. Topic models are considered as a tool for exploratory and predictive analysis of text. The most used topic model is the latent Dirichlet allocation (LDA) [2]. It posits that a small number of distributions over words, called topics, can be used to explain the observed data.

It is in this perspective that a new model that extends the LDA for the analysis of the personalized search problem was proposed in [5]. A user/topic distribution was added in the graphical model of LDA, involving the user in the generative process. The experimental results were not satisfactory and have not allowed to conclude that personalization increases the performance. The authors hypothesized that this negative effect on the ranking lists, may be related to the integration of the user in the generative process, because it makes the user very influential in the model and can be overwhelming information derived from data, while this information can be more useful. Thereafter, in [4], a model 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 generative process, but subtly introduced as part of the ranking formula, which is used to rank products for a given query. The authors concluded that there is an improvement in performance compared to non-personalized models. We will compare this system, called the PTM, with our proposed model. Two main shortcomings of the PTM are (1) it assumes a fixed prespecified number of topics regardless of the data and (2) it is unable to deal with new users.

We thus propose a new model, which enables to overcome this limit. Indeed, our HpDP model, is an extension of the HDP [13]. It allows to automatically learn the number of topics from the data. Once the topics have converged, we will be able to identify their number, and then introduce a user/topic distribution to determine the topical interests of users, and predict products for new users. We demonstrate the effectiveness of our approach through experiments conducted on web user sessions collected by a real online products comparator.

2 The HpDP model

2.1 Background

Mixture models explicitly model the existence of K sub-populations in the data. Each sub-population is represented by a probability distribution:

$$p(\boldsymbol{w}|\theta, \phi_{1:K}) = \sum_{k=1}^{K} \theta_k f(\boldsymbol{w}|\phi_k)$$

where w is a data point, θ_k is the mixture proportion and $f(.|\phi_k)$ is the density function of the sub-population k. Under a Bayesian setting, prior distributions are specified for θ and ϕ_k. Since they are multinomial distributions, we use the Dirichlet distributions as their conjugate priors. Given the specification of the prior distributions, Bayesian mixture models specify likelihood of data point w as follows:

$$p(\boldsymbol{w}|\Delta) = \int_{\theta} \int_{\phi_{1:K}} \sum_{k=1}^{K} \theta_k f(\boldsymbol{w}|\phi_k) d\theta d\phi_{1:K}$$

where Δ is the set of hyperparameters used to specify the prior distributions for θ and ϕ_k. To derive the posterior distributions for θ and ϕ_k, we turn to approximate inference methods since exact inference is intractable. Thus, we use Gibbs sampling [9] by introducing a latent variable z_n for each data point w_n to specify which sub-populations or mixture component the data w_n belongs to. The distribution of w_n conditioned by the latent variable z_n can be expressed as:

$$p(\boldsymbol{w}|z_n = k, \Delta) = \int_{\phi_k} f(\boldsymbol{w}|\phi_k) d\phi_k$$

The limit of the mixture models introduced above is that it is necessary to specify in advance the number of sub-populations K. To overcome this limitation, we assume that K is infinite:

$$p(\boldsymbol{w}|\pi, \phi_k) = \sum_{k=1}^{\infty} \pi_k f(\boldsymbol{w}|\phi_k)$$

where θ is a draw from π. The next step is to evaluate the specifications of ϕ_k and the prior distribution over the mixture proportion π, which is infinite-dimensional. The theoretical basis of this approach is the hierarchical Dirichlet processes (Figure 1 Right). In fact, $DP(\gamma, H)$ is a distribution over a probability measure G_0. It is defined by 2 parameters: $\gamma > 0$ which is a concentration parameter and H which is a base measure used to generate the parameters ϕ_k of the sub-populations K. We note $G_0 \sim DP(\gamma, H)$.

2.2 Model Description

In this section, we present the generative process of our proposed model, the hierarchical personalized Dirichlet processes (HpDP) given in Figure 1 (Left).

Let w_{di} be the ith word token in the user's query which led to a click on product d, and z_{di} its chosen topic. The generative process of the HpDP follows the following steps:

1. $\pi|\gamma \sim \text{Beta}(1, \gamma)$
2. $z_{di}|\theta_d \sim \text{Multinomial}(\theta_d)$
3. $w_{di}|z_{di}, \phi_{z_{di}} \sim \text{Multinomial}(\phi_{z_{di}})$

   We place priors on the parameters θ_d and $\phi_{z_{di}}$:
4. $\phi_{z_{di}} \sim H$
5. $\theta_d|\alpha \sim \text{Dirichlet}(\alpha\pi)$

   After topics convergence, and for a fixed number of topics:
6. $u_{di}|z_{di}, \psi_{z_{di}} \sim \text{Multinomial}(\psi_{z_{di}})$
7. $\psi_{z_{di}}|\epsilon \sim \text{Dirichlet}(\epsilon)$

where π is the distribution over topics and H the distribution over the vocabulary (query items), α and γ are concentration parameters.

Since prior knowledge of the number of topics is difficult, we propose this model that can determine it automatically. In the HpDP, we have an infinite number of topics (θ_d and π are infinite-dimensional vectors), and we use a stick-breaking representation [12] for π: $\pi_k = \tilde{\pi}_k \prod_{l=1}^{k-1}(1 - \tilde{\pi}_l)$ for $k = 1, 2, ...$ where $\tilde{\pi}_l|\gamma \sim \text{Beta}(1, \gamma)$.

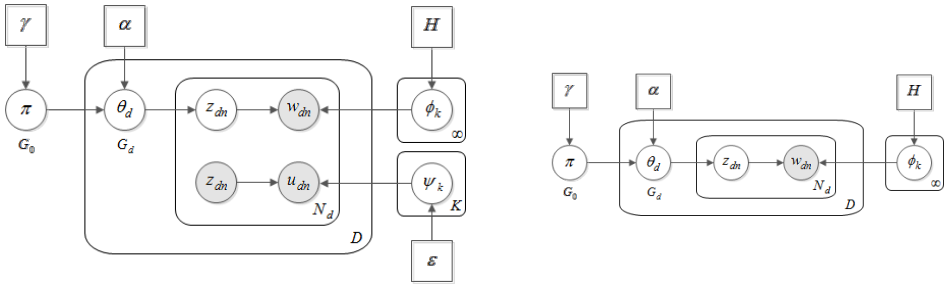

Fig. 1. (Left) Graphical Model of the HpDP, (Right) Graphical Model of the HDP

Using the notation of the Dirichlet process, we have: $G_d \sim DP(\alpha, G_0)$ and $G_0 \sim DP(\gamma, H)$ where: $G_d = \sum_{k=1}^{\infty} \theta_{dk}\delta_{\phi_k}$ and $G_0 = \sum_{k=1}^{\infty} \pi_k \delta_{\phi_k}$ are sums of point masses, and H is the base distribution.

2.3 Approximate Inference

We consider a product d with a probability distribution over words $z_{d1}, z_{d2}, ..., z_{dn_d}$ that make up the query that led to a click on product d. Since $G_d \sim DP(\alpha, G_0)$, we can characterize this distribution by describing how to generate $z_{d1}, z_{d2}, ..., z_{dn_d}$ using the Chinese Restaurant Process (CRP) [13]. In fact, the CRP considers n_d customers in a Chinese restaurant, with an unlimited number of tables. The first customer sits at the first table. The next customer sits at an occupied table with a probability proportional to the number of customers already present, or sits at an unoccupied table, with a probability proportional to α. Suppose customer i sits at table t_{di}, the conditional distributions are:

$$t_{di}|t_{d1}, ..., t_{di-1}, \alpha \sim \sum_t \frac{n_{dt}}{\sum_{t'} n_{dt'} + \alpha}\delta_t + \frac{\alpha}{\sum_{t'} n_{dt'} + \alpha}\delta_t^{new} \tag{1}$$

where n_{dt} is the number of customers currently at table t. When all customers have sat, we associate to table t a draw ζ_{dt} from G_0 and we set: $z_{di} = \zeta_{dt_{di}}$. We perform this process independently for each product d, we obtain all the G_d(s) together with an assignement of each z_{di} to a sample $\zeta_{dt_{di}}$ from G_0, with the partition structure given by CRP(s). We note that all ζ_{dt}(s) are i.i.d draws from $G_0 \sim DP(\gamma, H)$. We apply the same CRP partitioning process to the ζ_{dt}(s). Suppose that the customer associated with ζ_{dt} sits at table k_{dt}, the conditional distributions are:

$$k_{dt}|k_{11}, ..., k_{1n_1}, k_{21}, ..., k_{dt-1}, \gamma \sim \sum_k \frac{m_k}{\sum_{k'} m_{k'} + \gamma}\delta_k + \frac{\gamma}{\sum_{k'} m_{k'} + \gamma}\delta_k^{new} \tag{2}$$

Now, we associate with table k a draw ϕ_k from H and we set: $\zeta_{dt} = \phi_{k_{dt}}$. Thus, the generative process for the z_{di}(s) is completed, and we marginalize out G_0 and all the G_d(s). This generative process is called the Chinese Restaurant Franchise (CRF). The CRF is defined by three variables: $\boldsymbol{t} = (t_{di})$, $\boldsymbol{k} = (k_{dt})$ and $\boldsymbol{\phi} = \phi_k$. We describe an inference procedure based on Gibbs sampling $\boldsymbol{t}$, $\boldsymbol{k}$ and $\boldsymbol{\phi}$ given data points $\boldsymbol{w}$. Let $f(.|\phi)$ and h be the density funtions for $F(\phi)$

and H respectively. The conditional probability of t_{di} given the other variables is proportional to the product of a prior and likelihood term. The prior term is given by (1) and the likelihood is given by $f(w_{di}|\phi_{k_{dt}})$ where for $t = t^{new}$, we can sample $k_{dt^{new}}$ using (2), and $\phi_{k^{new}} \sim H$. Thus, the distribution is:

$$p(t_{di} = t|\boldsymbol{t}\backslash t_{di}, \boldsymbol{k}, \boldsymbol{\phi}, \boldsymbol{w}) \propto \begin{cases} \alpha f(w_{di}|\phi_{k_{dt}}) & if\ t = t^{new} \\ n_{dt}^{-i} f(w_{di}|\phi_{k_{dt}}) & if\ t \text{ currently used} \end{cases}$$

where n_{dt}^{-i} is the number of $t_{di'}$ equal to t except t_{di}.
In the same manner, the conditional distribution of k_{dt} is:

$$p(t_{dt} = k|\boldsymbol{t}, \boldsymbol{k}\backslash k_{dt}, \boldsymbol{\phi}, \boldsymbol{w}) \propto \begin{cases} \gamma \prod_{i:t_{di}=t} f(w_{di}|\phi_k) & if\ k = k^{new} \\ m_k^{-t} \prod_{i:t_{di}=t} f(w_{di}|\phi_k) & if\ k \text{ currently used} \end{cases}$$

where m_k^{-t} is the number of $k_{dt'}$ equal to k except k_{dt}.
Finally, the conditional distribution for ϕ_k is:

$$p(\phi_k|\boldsymbol{t}, \boldsymbol{k}, \boldsymbol{\phi}\backslash\phi_k, \boldsymbol{w}) \propto h(\phi_k) \prod_{di:k_{dt_{di}}=k} f(w_{di}|\phi_k)$$

For further details on the calculations, see [13].
Once topics have converged, we calculate the user/topic distribution ψ in order to consider the user profiles when ranking online products.

2.4 Calculation of the user/topic distribution

Since we are in the case where the variables are observed (topics which have converged in addition to users), we use the maximum likelihood method to estimate ψ (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 ϵ. Thus, for K topics and U users, L becomes:

$$L = \prod_{k=1}^{K}\prod_{u=1}^{U} \psi_{uk}^{N_{uk}} \prod_{k=1}^{K}\prod_{u=1}^{U} \frac{\Gamma(U\epsilon)}{\Gamma(\epsilon)^U} \psi_{uk}^{\epsilon_k - 1} = \frac{\Gamma(U\epsilon)}{\Gamma(\epsilon)^U} \prod_{k=1}^{K}\prod_{u=1}^{U} \psi_{uk}^{N_{uk}+\epsilon_k-1}$$

where Γ is the gamma function. Taking the logarithm of that term, we obtain:

$$\log L = \log \frac{\Gamma(U\epsilon)}{\Gamma(\epsilon)^U} + \sum_{k=1}^{K}\sum_{u=1}^{U} (N_{uk} + \epsilon_k - 1) \log \psi_{uk} \tag{3}$$

To simplify the calculations, we assume that the Dirichlet coefficients are equal: $\epsilon_1 = \epsilon_2 = ... = \epsilon_K = \epsilon$. We know that: $\sum_{u=1}^{U} \psi_{uk} = 1$, thereby: $\psi_{Uk} = 1 - \sum_{u=1}^{U-1} \psi_{uk}$. By injecting the last two equations in equation (3), we obtain:

$$\log L = \log \frac{\Gamma(U\epsilon)}{\Gamma(\epsilon)^U} + \sum_{k=1}^{K}\Big(\sum_{u=1}^{U-1} (N_{uk} + \epsilon - 1) \log \psi_{uk} + (N_{Uk} + \epsilon - 1) \log(1 - \sum_{u=1}^{U-1} \psi_{uk})\Big)$$

By taking the derivative of this term with respect to ψ_{uk}, we get:

$$\frac{\partial \log L}{\partial \psi_{uk}} = \frac{N_{uk} + \epsilon - 1}{\psi_{uk}} - \frac{N_{Uk} + \epsilon - 1}{1 - \sum_{u=1}^{U-1} \psi_{uk}} = \frac{N_{uk} + \epsilon - 1}{\psi_{uk}} - \frac{N_{Uk} + \epsilon - 1}{\psi_{Uk}}$$

By seting this term to zero, we get the maximum of ψ_{uk} that we denote $\widehat{\psi}_{uk}$:

$$\frac{N_{1k}+\epsilon-1}{\widehat{\psi}_{1k}} = \frac{N_{2k}+\epsilon-1}{\widehat{\psi}_{2k}} = ... = \frac{N_{Uk}+\epsilon-1}{\widehat{\psi}_{Uk}} = \frac{\sum_{u=1}^{U}(N_{uk}+\epsilon-1)}{\sum_{u=1}^{U}\widehat{\psi}_{uk}} = \sum_{u=1}^{U}(N_{uk}+\epsilon-1)$$

Thus: $\frac{N_{uk}+\epsilon-1}{\widehat{\psi}_{uk}} = \sum_{u=1}^{U}(N_{uk}+\epsilon-1)$
Finally, we get the expression of the user/topic distribution:

$$\widehat{\psi}_{uk} = \frac{N_{uk}+\epsilon-1}{\sum_{u=1}^{U}(N_{uk}+\epsilon-1)} \tag{4}$$

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

2.5 Predicting products 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 items (vocabulary containing words composing all users queries) using LDA.
2. Calculate the probability distribution of old users over the query items (the same vocabulary size).
3. Calculate the KL divergence between a new user distribution over query items and each of old users distributions.
4. Select the old user u^{old} for which the KL divergence is the lowest.
5. Predict products for the new user using his/her query and the user/topic distribution of the selected u^{old}.

2.6 Ranking online products

In this section, we describe formulas for ranking products using the parameters that were estimated based on the HpDP. We aim to return to the user a ranked set of products ($d \in \mathcal{D}$) according to their likelihood given his/her query $q = \{w_1, w_2, ..., w_n\}$. The formula in the case of a non-personalized model (LDA) is:

$$p(d|q) \propto p(d)p(q|d) = p(d)\prod_{w\in q} p(w|d) = p(d)\prod_{w\in q}\sum_{z} p(w|z)p(z|d) \tag{5}$$

where: $p(d) = \frac{N_d}{N}$, N_d is the number of words composing the user's query, which led to a click on product d and N is the total number of words composing all users'queries.
The ranking formula consists of multiplying a prior on the probability of the product (which we denote $p(d)$) with the probability of the query given the product (which we denote $p(q|d)$). This latter quantity can be estimated by introducing latent topics. Indeed, topic models allow to estimate the probability of words given topics $p(w|z)$ and the probability of topics given products $p(z|d)$. By introducing the user in the graphical model, we have information about the queries issued by a user. Thus, the user's preferences can be included into the

ranking formula. This means that we rank products according to their likelihood given both the query and the user as follows:

$$p(d|q,u) \propto p(d) \prod_{w \in q} p(w,u|d) = p(d) \prod_{w \in q} \sum_{z} p(w|z)p(u|z)p(z|d)$$

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

$$\tilde{p}(d|q,u) = p(d) \prod_{w \in q} \sum_{z} p(w|z)p(u|z)^{\lambda}p(z|d) \tag{6}$$

The introduction of this new parameter enables us to control the amount of influence that the user's topical interests may have on the ranking.

3 Experiments

3.1 Dataset

The dataset is from the query logs of a real products comparator[3] that connects potential buyers with major brands and distribution networks in the market of mobile telephony. We used two datasets, each one is based on a 1-month web log file. We have chosen to use data covering different periods to ensure that the model works regardless of the circumstances (promotion, flash sales, seasonal products, ...). The training data is generated automatically from log file without any human intervention.
For data cleaning, we have kept the queries which had resulted in a product selection. Then, we have selected only products for which more than 6 users had clicked on at least once. Finally, we selected only users with more than 6 remaining queries. This preprocessing step is carried out to ensure that users have made a significant number of queries and that products were also viewed reasonably. Table 1 gives a description of final corpus. 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 query has been made. Table 2 shows an example of two transactions from this query log. In our experiments, we consider that a product is represented by the triplet: (Package, Mobile Provider, Package features).

Dataset 1		Dataset 2	
Training subset size	1,053	Training subset size	1,049
Testing subset size	60	Testing subset size	65
# Users	130	# Users	132
# Products	103	# Products	106
# Query items	100	# Query items	101

Table 1. Datasets features.

[3] http://www.choisirsonforfait.com/

Id	Session	Package	Mobile Provider	Features	Date	User's query
3	73f08e	Mobile plan 1	Mobile Provider *A*	2-years contract	2013-01-15 11:57:22	sms & cell phone
2	ce77d6	Mobile plan 2	Mobile Provider *B*	2 hours plan	2013-01-15 11:57:15	1 hour of calls

Table 2. Log file format.

3.2 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. Thus, the training and testing subsets follow the same chronological order. We ranked products according to scores values defined above. Concerning the parameter setting, we set the Dirichlet prior α to be $0.1/K$, where K is the number of topics used for experiments. We evaluate the rankings by calculating two standard measures in the field of information retrieval: the Mean Reciprocal Rank (MRR) and the Mean Average Precision (MAP). We report these measures up to rank 6, since in information retrieval, it is valuable that pertinent products appear early in the ranked list. We consider that a ranked product is relevant if it is the same product the user had actually viewed. In order to determine if the hierarchical process is improving the ranking performance, we report another metric that we call the hierarchical personalization gain (HP-Gain). This metric compares the number of times the HpDP 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:

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

When the value of this metric is 0, then there is no change between the HpDP 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.

3.3 Results

Top K products-based evaluation Table 3 shows the results of the ranking experiments for the HpDP, the PTM and the LDA. An advantage key of our approach is that we do not have to vary the number of topics in order to obtain the optimum number of topics, since the HpDP enables to automatically determine them. In fact, we found 7 topics for the first dataset and 9 topics for the second one. However, to be fair with the two other models, we performed them by varying the number of topics. We notice an improvement compared to the PTM and the LDA. Moreover, 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 proposed product to the user is relevant, 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 HpDP is 69.47%, which means that the product it proposes to the user is broadly either ranked first or second. Therefore, we compute

another metric which is *Precision@n* to determine how much products should be proposed to the user so that the ranking will be the best.

		Number of Topics										
Measures	**Models**	5	7	10	15	20	25	30	35	40	45	50
MRR (%)	HpDP	-	**69.47**	-	-	-	-	-	-	-	-	-
	PTM	65.17	63.13	61.79	61.42	55.70	57.22	56.11	62.17	59.70	60.69	61.63
	LDA	61.08	59.79	55.71	53.35	61.63	59.21	57.53	59.71	58.93	57.56	59.09
MAP (%)	HpDP	-	**64.15**	-	-	-	-	-	-	-	-	-
	PTM	57.96	58.18	58.65	55.49	53.85	52.11	55.55	54.00	54.46	55.54	53.01
	LDA	56.97	55.12	53.61	51.03	60.37	55.47	54.24	54.51	52.23	51.24	52.37
		Number of Topics										
Measures	**Models**	5	9	10	15	20	25	30	35	40	45	50
MRR (%)	HpDP	-	**68.04**	-	-	-	-	-	-	-	-	-
	PTM	60.28	60.73	61.67	58.53	62.78	57.07	60.58	58.39	57.64	61.45	60.05
	LDA	55.15	56.12	56.39	57.81	60.13	58.00	54.59	57.25	55.96	58.86	57.73
MAP (%)	HpDP	-	**64.10**	-	-	-	-	-	-	-	-	-
	PTM	53.56	55.53	59.84	54.30	58.89	54.59	56.49	56.53	54.80	57.43	54.00
	LDA	53.72	53.91	52.52	53.07	57.15	54.05	56.06	51.46	50.67	54.53	52.79

Table 3. Ranking performance of the models on the test set over all queries ($\lambda = 0.10$): (Top) Results for the first dataset, (Bottom) Results for the second dataset.

	HpDP		PTM		LDA	
Measures	Dataset 1	Dataset 2	Dataset 1	Dataset 2	Dataset 1	Dataset 2
p@1	68.33 %	69.23 %	68.33 %	66.67 %	63.33 %	64.62 %
p@2	66.66 %	63.07 %	63.33 %	63.33 %	60.00 %	61.54 %
p@3	61.66 %	58.46 %	53.33 %	51.67 %	41.66 %	50.77 %

Table 4. Precision evolution according to the number of proposed products.

Table 4 shows the obtained result, which confirms our intuition about the relevance of the first and second ranked products, that are proposed to the user given his query. For the next experimentations, we will use 10 topics for PTM and 20 topics for LDA since their precisions are the best using these numbers of topics.

Influence of λ on the gain In this section, we highlight the influence of the parameter λ in terms of HP-Gain and hence on performance improvement. In fact, the parameter λ plays an important role in the ranking formula for the HpDP, since it enables 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 $\{0, 0.05, 0.1, ..., 0.3\}$. When $\lambda = 0$, the estimates of HpDP are the same as those given by the HDP. Figure 2 shows an improvement performance, over all queries. The HP-Gain varies between 14% and 28%. We chose to perform our experiments using $\lambda = 0.10$ since for this value, inter alia, the gain is maximum.

Influence of the click entropy on the gain When a given user/query pair had been observed before, we can use this information about prior clicks by

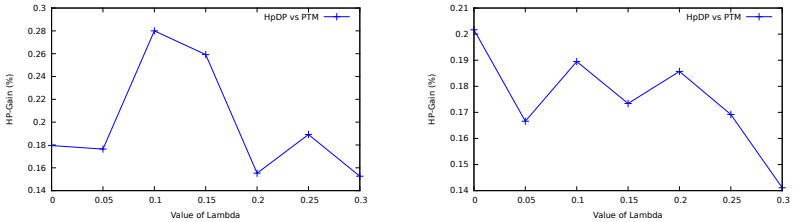

Fig. 2. The effect of varying the λ parameter in the ranking algorithm: (Left) Results for the first dataset, (Right) Results for the second dataset.

assuming that the user will again click on the same products as before. However, in almost cases, the user/query pair will be novel and we will not have such prior information to exploit. We will use 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 clicked products given the query q and $p(d|q)$ is the probability of selecting product d given the query q. Since entropy values vary in the range zero to the logarithm of the number of distinct products clicked on for a query, then, 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:

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

We calculated this measure for all queries. We separated these 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 HP-Gain for each of the two groups containing test queries. Figure 3 shows how the performance of the HpDP changes as the normalized click entropy of the queries evolves.

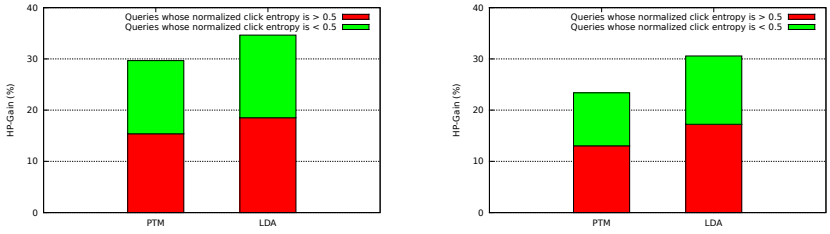

Fig. 3. The effect of query ambiguity in the ranking algorithm: (Left) Results for the first dataset, (Right) Results for the second dataset.

We notice that the HP-Gain increases as the click entropy increases. In fact, it reaches 18% for queries, the normalized click entropy of which is greater than 0.5 and it drops to 14% for queries, the normalized click entropy of which is lower than 0.5.

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 2.5 and we compare HpDP to LDA (PTM can not perform this task). HpDP found 7 topics for the first dataset and 11 topics for the second dataset. Again, we compute the MRR and MAP for LDA by varying the number of topics. Table 5 shows the obtained results. We notice that the MAP obtained using HpDP is always greater that the LDA's. Otherwise, the MRR obtained using HpDP outperforms LDA's except when considering 20 topics for LDA. Thus, we will use 20 topics for LDA when evaluating the *Precision@n*, given in Table 6. We notice again a performance improvement using our approach.

		Number of Topics										
Measures	**Models**	5	7	10	15	20	25	30	35	40	45	50
MRR (%)	HpDP	-	**64.99**	-	-	-	-	-	-	-	-	-
	LDA	61.23	61.84	63.35	61.58	**65.91**	62.29	62.21	57.50	61.37	62.38	60.25
MAP (%)	HpDP	-	**61.00**	-	-	-	-	-	-	-	-	-
	LDA	54.59	54.91	55.28	55.72	59.12	57.26	60.35	56.64	54.81	60.43	55.07

		Number of Topics										
Measures	**Models**	5	10	11	15	20	25	30	35	40	45	50
MRR (%)	HpDP	-	-	**63.90**	-	-	-	-	-	-	-	-
	LDA	59.07	54.41	55.78	58.14	**64.12**	53.90	59.98	60.69	59.74	58.06	62.93
MAP (%)	HpDP	-	-	**59.11**	-	-	-	-	-	-	-	-
	LDA	53.71	47.92	50.61	55.65	58.63	52.96	54.82	57.10	56.98	54.23	58.78

Table 5. Ranking performance of the models on the test set over all queries ($\lambda = 0.10$): (Top) Results for the first dataset, (Bottom) Results for the second dataset.

	HpDP		LDA	
Measures	Dataset 1	Dataset 2	Dataset 1	Dataset 2
p@1	65.57 %	67.14 %	63.90 %	65.71 %
p@2	62.30 %	62.86 %	60.66 %	58.57 %
p@3	59.02 %	60.00 %	57.37 %	57.14 %

Table 6. Precision evolution according to the number of proposed products.

4 Conclusion

In this paper, we have proposed a nonparametric Bayesian model that builds user profiles for personalized search. The comparison with other approaches indicated that performance can be improved through personalization. We also addressed the prediction task for new users. The obtained results showed that our model can further improve ranked lists

In our future work, we plan to analyze other families of function, that allow to control the influence of the user's topical interests on the ranking, in order to improve the gain. In addition, we intend to introduce dynamics in our model, either under Markovian or non-Markovian fashion.

Acknowledgments

This work was supported by Marketshot.

References

1. Bennett, P.N., White, R.W., Chu, W., Dumais, S.T., Bailey, P., Borisyuk, F., and Cui, X., *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), 2012.
2. Blei, D., Ng, A., Jordan, M.I., and Lafferty, J., *Latent Dirichlet allocation*. Journal of Machine Learning Research, 2003, 3:993-1022.
3. Chirita, P.A., Nejdl, W., Paiu, R., and Kohlschutter, C., *Using odp metadata to personalize search*. In Proceedings of the 28th Annual International Conference on Research and Development in Information Retrieval, (SIGIR), 2005.
4. Harvey, M., Crestani, F., and Carman, M., *Building user profiles from topic models for personalised search*. In Proceedings of the 22nd ACM International Conference on Information and Knowledge Management, (CIKM), 2013.
5. Harvey, M., Ruthven, I., and Carman, M., *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), 2011.
6. Matthijs, N., and Radlinski, F., *Personalizing web search using long term browsing history*. In Proceedings of the Fourth International Conference on Web Search and Data Mining, (WSDM), 2011.
7. Pretschner, A., and Gauch, S., *Ontology based personalized search*. In Proceeding of the International Conference on Tools with Artificial Intelligence, (ICTAI), 1999.
8. Qiu, F., and Cho, J., *Automatic identification of user interest for personalized search*. In Proceedings of the 15th International Conference on World Wide Web, (WWW), 2006.
9. Steyvers, M., and Griffiths, T., *Probabilistic Topic Models*. In T. Landauer, D. Mcnamara, S. Dennis, W. Kintsch. Latent Semantic Analysis: A Road to Meaning. Laurence Erlbaum, 2007.
10. White, R.W., Bailey, P., and Chen, L. *Predicting user interests from contextual information*. In Proceedings of the 32nd International Conference on Research and Development in Information Retrieval, (SIGIR), 2009.
11. Nguyen, T., Phung, D., Gupta, S., and Venkatesh, S., *Extraction of Latent Patterns and Contexts from Social Honest Signals Using Hierarchical Dirichlet Processes*. In the IEEE International Conference on Pervasive Computing and Communications (PerCom), 2013.
12. Sethuraman, J., *A Constructive Definition of Dirichlet Priors*. Statistica Sinica, 4, 1994.
13. Teh, Y., Jordan, M., Beal, M., and Blei, D., *Sharing Clusters Among Related Groups: Hierarchical Dirichlet Processes*. In Neural Information Processing Systems 17 (NIPS), 2005.