Personalized Search via Cluster Sensitive Ranking

Lai Lai Win
Computer University (Meiktila)
malei064@gmail.com

Abstract

Web search has become amazingly powerful in its ability to discover and exploit nearly any kind of information that comprises the web. However, as powerful and large as current web search engine are , stilled limited in their ability to always deliver key services to their users especially when there is a considerable number of user with different search intentions and needs. In this paper, we study both analytically and empirically personalized search emphasizing their retrieval aspects. We also propose an analytical model for personalized search unifying four critical of the problem namely link structure, document content, user queries and user preference.

1. Introduction

Currently, almost any search engine faces the increasingly difficult challenge of collecting, storing, processing, retrieving and distributing web data with different search intentions, needs and backgrounds. The goal of web search personalization is to allow a user to perform web search according to user preference or context [10]. In this paper, we conduct a theoretical study on the personalization aspect of web search by proposing an analytical intuitive model, motivated by [12]. Nowadays, allowing a user to perform web search according to their preference is an increasingly important web mining problem. This problem has two main parts :(1)how to represent a user's preference (2) how to use this information in search. Our propose model for personalized search first assumes that existence of underlying cluster structure for the target corpus in which we are going to perform the personalized search.

Furthermore, the analytical nature our model partially solves the ambiguity around what a personalized web search is. Our model exploits the linkage relations between underlying clusters taking into account content associated with each cluster. In this way, we aim to capture four critical aspects of namely, link structure, content generation, user query generation and user preference generation. Based on this unified model, we propose a spectral algorithm to perform personalized web search over any real web services that allows cluster structures showing the optimality of the proposed algorithm.

In the rest of this paper, section 2 describes related works. Section 3 provides an overview of the propose analytical intuitive model. Finally section 4 covers analysis of cluster-sensitive ranking algorithm and section 5 describes conclusion and future works.

2. Related Works

In this section, we review previous work related to the personalized ranking of web pages. Chirita et al. [6], propose a way of performing the web search using the ODP (Open Directory Project) Meta data. First user has to specify his/her search preference by selecting the set of topic that user is interested from ODP. Then at run time, the web pages returned by the ordinary search engine can be resorted according to distance between the URL page and user profile. Among different possible combinations of links analysis and their personalization methods, they reported that when the results produced from their distance function based personalization are combined with the page Rank algorithm, then they obtained the best search quality. Tee van et al. [10] study impact of various corpus, user, document and query representations on the personalization of web search. By modifying the BM25[8] which ranks documents based on their probability of the relevancy and the irrelevancy of retuned document from the ordinary search engine is assessed through the log sum over query terms occurring in the document. They report that the best combination features were found to be corpus Representation, user Representation, Document and Query Representation.

The our propose model in this paper is similar to but distinct from Haveliwala's incorporation of topic sensitivity into Page Rank[9] , which explored pre-computing offline rankings of each web page to be applied as biases during query processing. Such an a priori classification of topics is highly efficient, but we wish to be able to generate communications for arbitrary and yet possibly very specific ad-hoc collections.

3. Overview of Propose Model

One important objective of our personalized search framework is to find a model, general enough, to cover many real application scenarios. We achieve this goal by assuming that the targeted web service to be personalized has underlying cluster structures.

Given a set of clusters over the intended documents in which we want to perform personalized search, our model assumes that user preference is represented as a preference vector over these clusters.

When such direct gathering of user's search preference is not possible, we assume that the user would simply express their preferences over cluster structures. Next, we follow the approach taken by Achlioptas et al. [3] by proposing a model that considers linkage structure and content generation of cluster structures to produce a ranking of underlying clusters with respect to a user's given search query and preference. The rank of each document is obtained through the relation of a given document with respect to clusters and their respective ranking.

3.1. Cluster-Sensitive Page Ranking

Our proposed model is an extension of the model introduced by Achlioptas et al. [3].We start describing our model with a single set of clusters for the targeted corpus. Later, we extend it for multiple sets of clusters when various search features are considered. Let {C_1, Cm} be a clustering (not necessarily a partition) of size m for targeted corpus. We assume that there is an

N × m matrix Z whose (i, j) entry indicates whether or not page is part of cluster j.

Next, we assume that there exists a set of k unknown (latent) basic concepts whose combinations represent every topic of the web. Given such a set of k concepts, a topic is a k-dimensional vector λ, describing the contribution of each of the basic concepts to this topic; the ratio between the i-th and j-th coordinates of λ reflects the relative contributions of the underlying i-th and j-th concepts to this subject.

3.1.1. Authority and Hub for cluster

Based on notion of page's hub and authority values, we now introduce the concept of hub and authority values for clusters. With each cluster given cluster $C_p \epsilon C$, we associate two vectors.

- The first vector associated with C_p is a k-tuple $\tilde{A}^{(p)}$ whose each entry expresses the expected authority value that is accumulated in cluster C_p with respect to each concept. We define $\tilde{A}^{(p)}$ as $\tilde{A}^{(p)}(c)=\sum_{i \,\epsilon\, C_p} A(i,c)$ where A(i,c) is document i's authority value with respect to the concept c.
- The second vector associated with C_p is a k-tuple $\tilde{H}^{(p)}$ whose each entry expresses the expected hub value that is accumulated in cluster C_p with respect to each concept. We define $\tilde{H}^{(p)}$ as $\tilde{H}^{(p)}(c) = \sum_{i \,\epsilon\, C_p} H(i, c)$ is document i's hub value with respect to the concept c.

3.1.2. Link Generation over Cluster

Given cluster $C_p \epsilon$ C and cluster $C_q \epsilon$ C, our model assumes that the total number of links from pages in C_p to pages in C_q is a random variable with expected value equal to < $\tilde{H}^{(p)}$, $\tilde{A}^{(q)}$ >. Therefore, the link generation model among different clusters is described in terms of an m × m matrix $\tilde{W} = \tilde{H} . \tilde{A}^T$ where the p-th row of $\tilde{H}$ is $(H^{(p)})^T$ and the $\bar{q}$ Th row of A is $(A^{(q)})^T$. Each entry (p, q) of $\tilde{W}$ represents the expected number of links from C_p to C_Q.

Let $\widehat{W}$ be the actual link structure of documents for targeted corpus. We instantiate our link generation model of clusters described by $\overline{W}$ through $W=Z^T \widehat{W} Z$

3.1.3. Term Content Generation over Cluster

- For the first distribution, expresses the expected number of occurrence of terms as authoritative terms within all documents. More precisely, we assume the existence of a k-tuple S (u) A whose i-th entry describes the expected number of occurrences of the term u in the set of all pure authority documents in the concept i.
- For the second distribution, expresses the expected number of occurrences of terms as hub terms within all documents. More precisely, we assume the existence of a k-tuple S(u)H whose i-th entry describes the expected number of occurrences of the term u in the set of all pure hub documents in the concept i.

The Two distributions can be expressed in terms of two matrices, manly $\tilde{S}$ A, the l×k matrix, whose rows are indexed by terms, where rows u is the vectors ($\tilde{S}$ (u) H) T. Our model assumes that terms within cluster C_p having authority value $\tilde{A}$(p) and hub value $\tilde{H}$ (p) are generated from a distribution of bounded range where the expected number of occurrences of term u is

<Ã $^{(p),}$ $\tilde{S}$ (u) A> + < $\tilde{H}$ (p), $\tilde{S}$ (u) H>

We describe the term generation model of clusters with an m by l matrix $\tilde{S}$, where m is the number of underlying clusters and l is the total number of possible terms,

$$\tilde{S} = \tilde{H} . \tilde{S} \text{ TH} + \tilde{A} . \tilde{S}^{T}{}_{A}$$

The (i, j) entry in $\tilde{S}$ represents the expected number of occurrences of term j within all documents in cluster i. Let $\hat{S}$ be the actual term document matrix of all documents in the targeted corpus. Analogous to the previous link generation model of clusters, we instantiate our term generation model of clusters described by $\tilde{S}$ through $\bar{S} = \hat{S}$ .Z.

3.1.4. User Query

The query generation process in our model is given as follows:

- The user chooses the k-tuple υ describing the topic he wishes to search for in terms of the underlying k concepts.
- The user computes the vector $\tilde{q}^{T}$= $\tilde{\upsilon}^{T} \tilde{S}^{T}{}_{H}$ where the entry of $\tilde{q}$ is the expected number of occurrences of the term υ in a cluster.
- The user them decides whether or not to include term υ among his search terms by sampling from a distribution with expectation $\tilde{q}$ [υ]. We denote the instantiation of the random process by $\bar{q}$ [υ].

  The input to the search engine consists of the terms with non-zero coordinates in the vector $\bar{q}$.

3.2. Search Preference

For user preference, we consider slightly more general scenario in which the user expresses his search interests through a set of keywords (terms).More precisely, our user search preference is given by:

- The user expresses his search preference by providing a vector p! T over terms whose i-th entry indicates his/her degree of preference over the term i.
- Given the vector $\tilde{p}^{T}$, the preference vector clusters is obtained as $\tilde{p}^{T}.\tilde{S}^{T}$.

3.3. Final Ranking

Based on our previous model, we assess the authoritativeness of each cluster with retained aspect to a topic υ: the relative authoritativeness of two clusters Cp and C_q on a topic υ is given by the ratio between < υ, A $^{(p)}$)> and, < υ, $\tilde{A}^{(q)}$ >. When the user's preference is given, the relative authoritativeness of two clusters C_p and C_Q on the topic υ is given by the ratio between < υ, ($\tilde{p}^{T}$. $\tilde{S}^{T}$) $_{p}$. $\tilde{A}^{(p)}$>and< υ, ($\tilde{p}^{T}$. $\tilde{S}^{T}$) $_{q}$. $\tilde{A}^{(q)}$>. Cluster sensitive page ranking (with user's preference already integrated) is obtained by computing M= $\upsilon^{T} \tilde{A}^{T} \bar{p}^{T} \tilde{S}^{T} I_m G$ where I_m is the identity matrix of size m. Let μ (C_i, x.q) be cluster-sensitive page rank for page x. Since we already have assumed that we have a page ranking R(x, q) the final rank for page x (i.e personalized ranking) can be obtained as

PR(x)

$$= \sum_{Ci \in C} R(x,q).\mu(C_i, x, q) = R(x,q).\mu(C_s(x), x, q)$$ where C_s(x) is a cluster in C in which x ϵ C_s(x).

3.4. Algorithm for Cluster-Sensitive Page Ranking (CSPR)

Given our model that incorporates link structure, content generation, user preferences, and query, we rank clusters of documents using a spectral method. In contrast to the original SP algorithm which works at the document level, our algorithm works at the cluster level making our algorism computationally more attractive and consequently more practical. For our algorithm, in addition to the SVD computation of $\tilde{M}$ and $\tilde{W}$ matrices, the SVD computation of $\tilde{S}$ is also required.

3.4.1. Notation

For two matrices A and B with an equal number of rows, let [A|B] denote the matrix whose rows are the concatenations of the rows of A and B. Let $\sigma_i(A)$ denote the i-th largest singular value of a matrix A. Let $\tau_i(B) \geq 1$ denote the ratio between the primary singular value and the i-th singular value of $B : \tau_i(B) = \sigma_1(B) / \sigma_i(B)$. Note that $\tau_i(B) \geq 1$ and if $\tau_i(B)$ = 1 then this means that the singular value do not drop at all, the larger $\tau_i(B)$ is the larger the drop in singular values. Let [0^n] denote a row vector with larger $\tau_i(B)$ is the larger the drop in singular values. Let [0^n]denote a row vector with n zero, and let $[0^{i \times j}]$ denote an all zero matrix of dimensions i×j. We use a standard notation for the singular value decomposition (SVD) of a matrix. More precisely, given a matrix $B \in R^{n \times m}$, let the singular value decomposition (SVD) of B be $U \sum V^T$ where U is a matrix of dimensions n×rank (B) whose columns are orthonormal, ∑ is a diagonal matrix of dimensions rank (B) ×rank (B), and V^T is a matrix of dimensions rank (B) ×m whose rows are orthonormal. The (i, i) entry of ∑ is $\sigma_i(B)$.

3.4.2. Algorithm Description

The algorithm performs the following pre-processing of the entire corpus of documents, at which the search is performed, independently of the query.

Pre-processing Step

1. Let $\overline{M}$ = [$\overline{W}$ ^{T}| $\overline{S}$]. Recall that $\overline{M} \in R^{m \times (m+1)}$ (m is the number of clusters and l it's the number of terms). Compute the SVD of the matrix as

$$\vec{M} = U_{\overline{M}} \sum_{\overline{M}} VT_{\overline{M}}$$

2. Choose the largest index τ such that the difference $|\sigma_\tau(\overline{M}^*) - \sigma_{\tau+1}(\overline{M}^*)|$ is sufficiently large (we require $\omega(\sqrt{(m+1)})$. Let $\overline{M}_\tau^*$ = $((U_{\overline{M}})_\tau (\sum_{\overline{M}})_\tau (VT_{\overline{M}})_\tau$ be the rank τ -SVD approximation to $\overline{M}$.

3. Compute the SVD of the matrix $\overline{W}$ as

$$\overline{W}^* = U_{\overline{W}} \sum_{\overline{W}} VT_{\overline{W}}$$

4.Choose the largest index t such that the difference $|\sigma_t(\overline{W}^*) - \sigma_{t+1}(\overline{W}^*)|$ is sufficiently large (we require $\omega(\sqrt{(t)})$. Let $W_t = (U_{\overline{W}})_t (\sum_{\overline{W}})_t (VT_{\overline{W}})_t$ be the rank t-SVD approximation to $\overline{W}$.

Let a^T (s) denote the authority value of page s. The final rank of page s is simply computed as

a^T (s).R(s, q)

5.Compute the SVD of the matrix $\overline{S}$ as

$$\overline{S}^* = U_{\overline{S}} \sum_{\overline{S}} V_{\overline{S}}^t$$

6. Choose the largest index σ such that the difference $|\sigma_0(\overline{S}^*) - \sigma_{0+1}(\overline{S}^*)|$ is sufficiently large (we require $\omega(\sqrt{(0)}))$. Let $\overline{S}^* = (U_{\overline{S}})_0 (\sum_{\overline{S}})_0 (V_{\overline{S}}^T)_0$ be the rank 0-SVD approximation to $\overline{S}$.

Query Step

Once a query vector $\overline{q}^T \in R^l$ is presented, let $\overline{q}^T = [0^m \mid \overline{q}^T] \in R^{m+1}$. Then, we compute the vector

$$\omega^T = \overline{q}^{t^T} \overline{M}^*{}_\tau{}^{-1} \overline{W}^*{}_t . p^T . \overline{S}_o^{*T} I_m$$

Where $\overline{M}^*{}_\tau{}^{-1} = (VT_{\overline{M}})_\tau (\sum_{\overline{M}})_\tau^{-1} (U_{\overline{M}})_\tau$ is the pseudo-inverse of $\overline{M}_\tau$.

The authority value of cluster $C_{p \text{ is}}$ $\omega^T(p)$. to compute the authority value for each page, we compute the vector

$$a^T = \omega^T . Z$$

3.5.2. Rank Computation with respect to Multiple Clusters

We extend our model to compute cluster-sensitive ranking with respect to multiple cluster structure as follows. Let Ω = { Ω1,……,Ωt } be the set o multiple cluster structure of the targeted.

4. Analysis of Cluster-Sensitive Ranking Algorithm

The next theorem states about how well our cluster-sensitive ranking algorithm actually approximates our proposed model.

Theorem 1 Assume that the link structure for clusters, term content for clusters and search query are generated as described in our model: $\overline{W} = \tilde{H} \tilde{A}^T$, $\overline{S}$ is an instantiation of $\overline{S} = \tilde{A} \overline{S}^T A + \tilde{H} \overline{S}^T H$, $\overline{q}$ is an instantiation of $\tilde{q} = \upsilon^T \overline{S}^T H$. User's preference is provided by pT. Additionally, we have

1. $\overline{q}$ has ω (k.ϒ$_k$ ($\overline{W}$)^{2}ϒ$_{2k}$ ($\overline{M}$)^{2}ϒk(G^T)) terms.

2.σ$_k$ ($\overline{W}$) εω (ϒ$_{2k}$ ($\overline{M}$) ϒk (GT)$\sqrt{m}$) and $\sigma 2k$ $(\overline{M})$ϵω (ϒk ($\overline{W}$) ϒ$_{2k}$ $(\overline{M})$ ϒk (G^T) $\sqrt{m}$

3.$\overline{W}$, $\overline{HS}$ $^T{}_A$ and $\overline{S}$ $^T{}_H$ are rank k, $\overline{M}$ = [$\overline{W}$ ^{T}| $\overline{S}$] is rank 2k, l=O (m), and m=O (k)

Then the algorithm computes a vector of authorities that is very close to the correct ranking.

The following theorem is an analytical statement of a somewhat obvious fact. If we have two users in which one is very expressive in his/her search preference (e.g. his/her search preference is strongly biased toward certain clusters) while other one is less expressive in his/her search preference (e.g. his/her search preference is spread evenly across clusters), then there is a higher chance that the cluster-sensitive ranking produced by our algorithm is more strongly influenced by the former's search preference.

Theorem 2 Given a pair $(\upsilon^T \tilde{A}^T \tilde{p}^T \tilde{S}^T I_m)_{(i)}$, $(\upsilon^T \tilde{A}^T \tilde{p}^T \tilde{S}^T I_m)_{(j)}$, we say that pair is flipped if $(\upsilon^T \tilde{A}^T \tilde{p}^T \tilde{S}^T I_m)_{(i)} < (\upsilon^T \tilde{A}^T \tilde{p}^T \tilde{S}^T I_m)_{(j)}$ but $(\upsilon^T \tilde{A}^T{}_{(i)} > (\upsilon^T \tilde{A}^T)_{(j)}$ or $(\upsilon^T \tilde{A}^T \tilde{p}^T \tilde{S}^T I_m)_{(i)} > (\upsilon^T \tilde{A}^T \tilde{p}^T \tilde{S}^T I_m)_{(j)}$ but $(\upsilon^T \tilde{A}^T)_{(i)} < (\upsilon^T \tilde{A}^T)_{(j)}$. Let

$$m_{ij}(p) \equiv \frac{\min(p(C_i), p(C_j))}{\max(p(C_i), p(C_j))} \text{ if}$$

$p(C_i) \neq p(C_j)$ and $m_{ij}(p) \equiv 0$ if $p(C_i) = p(C_j)$.

Let F be the random variable for the total number of flipped pairs.

If

$$\sum_{i,j} m_{ij}(p) \leq \sum_{i,j} m_{ij}(p')$$

Then, we have

$$E\,[F]_p < E\,[F]_{p'}$$

Where $E\,[F]_p$ is the expected number of pairs which are flipped with the preference vector p.

Corollary 1 Suppose that the preference vector over the clusters is given as either ϵ or c + ϵ for each $p(C_i)$,i.e. the preference vector over the clusters is a weighted binary vector. Let ≠p be the total number of entries in $p(C_q)$ that are not ϵ. If

$$\#\mathrm{p} \leq \# p' \qquad (4.1)$$

$$\#\mathrm{p} + \# p' \leq l \qquad (4.2)$$

Then, we have

$$E\,[F]_p < E\,[F]_{p'}$$

5. Experimental Results

In our experimental we used "Regional / North America: United States" data sets from Dmoz as seed pages we ran a small scale crawler for 5 days. We first choose 11 samples keywords, 7 locations and then combined these keywords and locations to build the query sting. We also used MSN search engine. The total number of pages collected was around 665000. Cluster would correspond to a set of data items or web pages related to the specific geographic location.

To evaluate the quality of the results returned by each algorithm, we constructed a ground-truth set. We first merged all top 10 pages returned by each algorithm and those from MSN into one single set. We rate each page in the set as either "Relevant" or "Highly Relevant" by analyzing its content.

Day care, Financial Service, Fitness, Health, Shopping Seafood, Hotel, Italian Restaurant, Plumbing, Real Estate, School

Figure1. Query keywords used in Experiments

(Austin, TX), (Chicago, IL), (Houston, TX), (Miami, FL),(Los Angeles, CA),(New York, Ny), (Tucsty, AZ)

Figure2. Location used in experiments

HR: Definitely related to both query term as well as to the query's dominant location.

R: Probably related to either the query term or the query's dominant location.

Using this ground-truth set we assessed the quality of our ranking by comparing this set against the top 10 results returned by each algorithms. Once again, we used the precision over the top 10 as the measure for evaluation. In Table 1 we report the average HR and R ratio of all algorithms, propose algorithms(CSPR) and MSN results.

6. Conclusion and Future Works

In this paper, we started our study by proposing a way of modeling a personalized search scenario in which one is integrating the personalized search capability into already existing real web services.

Our model views a personalized search as the combination of a user's search preference, user's query, classical ranking of pages, and ranking of pages with respect to a given clustering. We propose an algorithm to compute the personalized ranking for our model. Thus, our main contribution here was that any web service whose underlying service architecture was structured through a set of either explicit or implicit clusters could be easily equipped with a personalized search capability.

Currently, there is a plethora of works on web personalization in both industry and academia. However, there is no way of assessing how proposed personalized algorithms or services are different from each other in a spirit similar to that of [2, 11]. Therefore, we plan to extend our model to study different personalized methods within one single framework. Additionally, we plan to propose some axiomatic approaches for personalized web search motivated by recent works on personalized Database systems [4, 7] and axiomatization of web ranking function [1].

References

[1] Altman and m. Tennenholtz."Ranking Systems: the page rank axioms", *In Ec'05: Proceedings of the 6th ACM conference on Electronic commerce,* Pages 1.8, New York, NY, USA, 2005.ACM press.

[2] A.Borgatti, G.O .Robert, J.S Rosential and P.Tsaparas."Finding authorities and hubs from link structures on the world wide web". In Proceeding of www 2001, pages 415-429 ACM, May 2011

[3] A. Fiat, A.R.karlin, D. Achilioptas and F.Mcsherry "Web Search via Hub synthesis" .In focs, page 500-509 ACM.2001

[4] G.koutrika and Y.E.Ioannidis "Personalized Queries under a Generalized Perference model" .In *ICDE,* 2005.

[5] M. Aktas, M.Nacar and F.Menczer."Personalizing page rank based on domain profiles". *In Web KDD 2004 ACM,* Augest 2004

[6] P.A. Chirita, W.Nejdl, R.Paiu and C.Kohlschuetter."Using Odp metadata to personalize search" .In *SIGIR* 2005.ACM, August 2005

[7] R. Fagin, P.G Kolaitis ,R.J .Miller and L.Popa. "Data exchange: semantics and query answering". *Theor.Comput .Sci*, 336(1):89-124 2005.

[8] S.Jones, K.Walker and S.Robertson ."A probabilistic model of information retrieval development and status". *Technical Report Technical Repor*t TR-446, Cambridge University Computer Laboratory, 1998

[9] T.Haveliwala," Efficient computation of page rank". *Technical report,* Stanford University, September 1999.

[10] J.Teevan, S, T.Dumaisand E.Horvits."Personalizing search via automated analysis of interests and activities". In *SIGIR,* pages 449-456, 2005

[11] Y.Li.Y.Lu, L Wanh,"Detecting dominant locations from search queries". In *SIGIR*, pages 424-431 2006