Improving Search Personalisation with Dynamic Group Formation

Thanh Vu[1], Dawei Song[1,3], Alistair Willis[1], Son N. Tran[2], and Jingfei Li[3]
The Open University, Milton Keynes, UK[1] City University London, London, UK[2] Tianjin University, Tianjin, P.R.C[3]
{thanh.vu, dawei.song, alistair.willis}@open.ac.uk, son.tran.1@city.ac.uk, tjucsljf@foxmail.com

ABSTRACT

Recent research has shown that the performance of search engines can be improved by enriching a user's personal profile with information about other users with shared interests. In the existing approaches, groups of similar users are often statically determined, e.g., based on the common documents that users clicked. However, these static grouping methods are query-independent and neglect the fact that users in a group may have different interests with respect to different topics. In this paper, we argue that common interest groups should be dynamically constructed in response to the user's input query. We propose a personalisation framework in which a user profile is enriched using information from other users dynamically grouped with respect to an input query. The experimental results on query logs from a major commercial web search engine demonstrate that our framework improves the performance of the web search engine and also achieves better performance than the static grouping method.

Categories and Subject Descriptors: H.3.3 [Information Systems Applications]: Information Search and Retrieval

Keywords: Search Personalisation; Latent Dirichlet Allocation; Query Log; Re-ranking

1. INTRODUCTION AND BACKGROUND

Recently, search personalisation has attracted increasing attention [1, 3, 5, 8, 9]. Unlike classical search methods, personalised search systems use personal data about a user to tailor search results to the specific user. This information can be considered as a *user profile*. A widely used type of user profiles represents the topical interests of the user [5, 7]. A typical approach is to build user profiles using the main topics discussed in the relevant documents [1, 9]. The topics of a document can be obtained from a human-generated online ontology, such as the Open Directory Project (ODP) [1, 9]. However, this approach suffers from a limitation that many documents may not appear in the online categorisation scheme. Moreover, it requires expensive manual effort to determine the correct categories for each document [5].

The effectiveness of a personalised search system relies on collecting user profiles that are rich enough [8]. Recent studies [3, 8, 9] show that a user profile can be enriched by using data from a group of users who share common interests, determined statically, e.g., based on the common documents that users clicked [3]. Despite being successful in improving search results, the static grouping methods neglect the fact that users in a group may have different interests with respect to different topics. In order to capture this characteristic, we argue that the grouping method needs to be dynamic and dependent on the input query (i.e., different input queries should return different groups of similar users).

In this paper, we propose a model for query-dependent user grouping and leveraging the dynamic group information to enrich user profiles for search personalisation. In Section 2.1, we construct user profiles based on relevant documents extracted from query logs over a topic space. We utilise a latent topic model to automatically derive topics instead of using a human-generated ontology [1, 9]. However, instead of assuming that all clicked document are relevant as in [3, 7], we use the Satisfied (SAT) Clicks criterion [4] to identify SAT clicked documents. In Section 2.2, we introduce a novel method to dynamically group users given the input query of the current user. In Section 2.3, we leverage the group data to personalise the search results of the current user. Sections 3 and 4 present the experimental setup and evaluation results.

2. PERSONALISATION FRAMEWORK

2.1 Constructing a User Profile

We employ the Latent Dirichlet Allocation (LDA) [2] to learn latent topics from the SAT clicked documents extracted from query logs. Let T, W and D be variables representing a latent topic, a word and a document respectively. The user variable is denoted as U. Let t, w, d and u denote instances of T, W, D and U respectively. After training the LDA model, we get two distributions $P(W|T)$ and $P(T|D)$. $P(W|T)$ defines a distribution of words for each topic showing how relevant they are to the topic. For example, a topic about football would give high probabilities to words like "score" and"goal", and low probabilities to "Windows" and "Linux". $P(T|D)$ defines a distribution of these latent topics for each document showing how relevant they are to the document. For example, a document about a football match

SIGIR'14, July 6–11, 2014, Gold Coast, Queensland, Australia.
Copyright 2014 ACM 978-1-4503-2257-7/14/07 ...$15.00.
http://dx.doi.org/10.1145/2600428.2609482.

would give a high probability to the topic about "football" and a low probability to the "OS" topic.

We define the interest of a user u on a latent topic t as a conditional probability:

$$p(t|u) = \frac{1}{N_{SAT(u)}} \sum_{d \in SAT(u)} p(t|d) \tag{1}$$

Here, $SAT(u)$ is a set of SAT clicked documents from user u, $N_{SAT(u)}$ is size of that set, and $p(t|d)$ is the probability of topic t given the document d. $P(T|U)$ shows how each user is interested in the latent topics contained in her/his SAT clicked documents.

2.2 Query-dependent User Grouping

We propose a query-dependent grouping method to group users who share common interests with the current user, with respect to the current query. In particular, given a user, different input queries could result in different user groups. Formally, for a user u and an input query q, we define the similarity function between u and another user v as the conditional probability $p(u,v|q)$. The top K-nearest users to u can be extracted by ranking all other users in a descending order of their similarity scores to u:

$$G(u,q) = \{v|rank(p(u,v|q)) \leq K\} \tag{2}$$

Here, $rank(p(u,v|q))$ is the rank of the similarity score $p(u,v|q)$ between u and v given a query q.

Now the key is to estimate $p(u,v|q)$. Applying the Bayes' Rule, we have:

$$p(u,v|q) \propto p(q|u,v)p(u,v) \tag{3}$$

where $p(u,v)$ is calculated as:

$$p(u,v) = \frac{N_{IN_{u,v}}}{\sum_{i,j} N_{IN_{i,j}}} \propto N_{IN_{u,v}} \tag{4}$$

Here, $IN_{u,v}$ is the intersection between SAT clicks of u and SAT clicks of v. $N_{IN_{u,v}}$ is the size of $IN_{u,v}$.

Next, to calculate $p(q|u,v)$, we apply the marginal probability over the latent topics T to receive:

$$p(q|u,v) = \sum_t p(q|u,v,t)p(t|u,v) \tag{5}$$

The probability $p(t|u,v)$ represents how likely that both users u and v are interested in the topic t. We define:

$$p(t|u,v) = \frac{1}{N_{IN_{u,v}}} \sum_{d \in IN_{u,v}} p(t|d) \tag{6}$$

Likewise, the probability $p(q|u,v,t)$ represents how likely that both users u and v are interested in issuing the query q given the topic t. Here, the query q is represented as a set of words $W_q = \{w|w \in q\}$. To calculate $p(q|u,v,t)$, we assume query words are mutually independent and also independent of u and v given the topic t. We have:

$$p(q|u,v,t) = \prod_{w \in W_q} p(w|t) \tag{7}$$

Applying Eq. (7) and Eq. (6) to Eq. (5), we have:

$$p(q|u,v) = \frac{1}{N_{IN_{u,v}}} \sum_t \prod_{w \in W_q} p(w|t) \sum_{d \in IN_{u,v}} p(t|d) \tag{8}$$

Finally, applying Eq. (4) and Eq. (8) to Eq. (3) we have:

$$p(u,v|q) \propto \sum_t \prod_{w \in W_q} p(w|t) \sum_{d \in IN_{u,v}} p(t|d) \tag{9}$$

Eq. (9) shows that the query-dependent similarity between two users depends on the common documents that they both have visited and how likely the current query relates to these documents through latent topics.

2.3 Personalising Search Results using Group Information

After obtaining the K-nearest users who share common interests with the user u given input query q ($G_{u,q}$, or in short G_u), we can leverage these users' profiles to enrich the current user's. We define $p^*(t|u)$ as group-enriched conditional distribution indicating the topic interests of u:

$$p^*(t|u) = \frac{p(t|u) + \sum_{v \in G_u} p(t|v)}{1+K} \tag{10}$$

We then utilise the enriched user profile to re-rank the original list of documents returned by the search engine. The detailed steps are as follows:

(1) Download the top n ranked search results (as recorded in query logs) from the search engine for a query. We denote a downloaded web page as d and its rank in the search result list as $r(d)$.

(2) Compute a personalised score for each web page d given the current user u as the probability $p(d|u)$:

$$p(d|u) \propto \sum_t p(d,u|t)p(t) \tag{11}$$

Given a topic t, we assume u and d are mutually independent; the prior $P(D)$ is a uniform distribution. Applying the Bayes' Rule, Eq. (11) becomes:

$$p(d|u) \propto \sum_t \frac{p(t|d)p(t|u)}{p(t)} \tag{12}$$

Here, $p(t|d)$ has been available as an output of the LDA inference process in Section 2.1. Next, to leverage the group information, in Eq. (12), we use $p^*(t|u)$ instead of $p(t|u)$:

$$p(d|u) \propto \sum_t \frac{p(t|d)p^*(t|u)}{p(t)} \tag{13}$$

(3) Combine the personalised score $p(d|u)$ and the original rank $r(d)$, to get a final score τ as:

$$\tau(u,d) = \frac{p(d|u)}{r(d)} \tag{14}$$

As we do not have access to the the original relevance score between an input query and a returned document given by the baseline search engine, we use $1/r(d)$ as an estimate. We combine $p(d|u)$ and $r(d)$, as they reflect different aspects in ranking documents.

3. EXPERIMENTAL METHODOLOGY

3.1 Evaluation Metrics

We use two measurement metrics to evaluate a personalised search approach, which are: inverse average rank and personalisation gain.

(1) Inverse Average Rank (IAR) The average rank (AR) over a set of test queries S [3] is defined as follows:

$$AvgRank = \frac{1}{|S|} \sum_{s \in S} \frac{1}{|P_s|} \sum_{p \in P_s} R(p) \tag{15}$$

where P_s is a set of relevance (SAT clicked) web pages for a test query s; $R(p)$ is the rank of a page p. A smaller AR indicates a better overall quality of the ranked results [3]. For ease of use, in this paper, we define an inverse AR (IAR) metric as:

$$IAR = \frac{1}{AvgRank} \tag{16}$$

The higher IAR score indicates the better ranking quality.

(2) Personalisation Gain (P-Gain) shows how stably the personalisation improves the ranking performance over a baseline across all test queries [5]. The metric compares the number of relevant web pages promoted to a higher rank against the number of relevant pages obtaining worse ranking after using the personalisation algorithm:

$$\text{P-Gain} = \frac{\#better - \#worse}{\#better + \#worse} \tag{17}$$

A higher positive P-Gain value indicates a better overall robustness of a personalisation algorithm in term of improving performance over the baseline.

3.2 Dataset And Evaluation Methodology

The dataset used in our experiments is a sample of query logs from a major commercial web search engine for 15 days from 1^{st} to 15^{th} July 2012. The query logs contain searching data from 106 anonymous users. A log entity consists of an anonymous user identifier, a query, top-10 returned URLs, and clicked results along with the user's dwell time.

For evaluation, we use the SAT criterion to identify the satisfied clicks from the query logs as either a click with a dwell time of at least 30 seconds or the last result click in a session. Then, we split the dataset into training and test sets. The training set contains the log data in the first 10 days and the test set contains the log data in the remaining 5 days. Table 1 shows the statistics. We also consider the SAT clicks as the ground truth of the test data. In our experiments, we evaluate our proposed method by comparing the original rank list given by the commercial search engine and the re-ranked list given by our methods with the evaluation metrics defined in Section 3.1.

Table 1: Basic statistics of the dataset

Item	ALL	Training	Test
#days	15	10	5
#users	106	106	106
#queries	17947	11695	6252
#distinct queries	8008	5237	3102
#clicks	24041	15688	8353
#SAT clicks	16166	10607	5559
#SAT clicks/#queries	0.9008	0.9069	0.8892

In addition to reporting the overall performance, we also analyse the results with respect to the concept of click entropy [3]. A smaller click entropy value indicates that more agreement between users on clicking a small number of web pages [3]. Dou et al. [3] also pointed out that if the click entropy is small, the personalisation process can even deteriorate the search performance. In the experimental data, about 80% queries have a low click entropy between 0 and 1; 15% queries have a click entropy between 1 and 2; and about 5% queries have a high click entropy (≥ 2).

4. EXPERIMENTAL RESULTS

In the experiment, we compare the performances of the baseline and three personalisation strategies, namely S_Profile, S_Group and D_Group. The baseline is the original ranked results from the web search engine; the S_Profile is a personalisation approach using the current user profile; the S_Group uses the profile enriched with information from static grouping $p(u, v)$; and the D_Group is enriched with information from dynamic grouping $p(u, v|q)$.

For training LDA model, we employ the Mallet implementation [6] of the LDA model. We also observe that the choice of hyper-parameter has little impact on overall performance. Therefore, in this work we set the number of topics as 100 and the hyper-parameters as in [2]. Since the number of the anonymous users is relatively small (106), we set the number of nearest neighbours K = 5 for both S_Group and D_Group.

4.1 Overall Performance

In this section, we analyse the experimental results of the personalised strategies in terms of IAR and P-Gain. Table 2 shows that all personalisation strategies can improve over the baseline (i.e., all reported changes are positive in IAR and P-Gain values). Interestingly, the D_Group method has the highest ($p \ll 0.01$ with the *paired t-test*) improvement of 8.12% over the baseline (using IAR metric). S_Group and S_Profile methods also have significant improvements of 7.64% and 5.94% respectively ($p \ll 0.01$) over the baseline. This shows that latent topic-based personalisation methods generally yield better web search performance.

Table 2: Overall performance of the methods.

Strategy	IAR	P-Gain
Baseline	*0.3473*	-
S_Profile	0.3679	0.1579
S_Group	0.3738	0.2848
D_Group	**0.3755**	**0.3253**

Table 2 also shows that the Group-based methods (S_Group and D_Group) improve the IAR and P-Gain values by at least 1.60% and 80.37% respectively over the S_Profile($p \ll 0.01$). Consistent with [8], our result confirms that the information from the group of users who share some common interests is helpful in building better user profiles.

Table 3: Numbers of better and worse ranks in comparison with the baseline and P-Gain

Strategy	#Better	#Worse	P-Gain
S_Profile	913	664	0.1579
S_Group	882	491	0.2848
D_Group	884	**450**	**0.3253**

In addition, the dynamic grouping method (D_Group) leads to improvements of 0.45% and 14.22% in IAR and P-Gain respectively over the static grouping method (S_Group) ($p \ll 0.01$). Furthermore, in Table 3, even though D_Group method leads to only few (2) more improved ranks than S_Group, the former has much (41) less worse ranks. This suggests that dynamic group information with respect to an input query could gain performance over static group information, especially for reducing the number of incorrect re-rankings.

Remark on efficiency issue: Both LDA model training and static group formation are done offline, and the dynamic group formation is done partially offline[1]. Thus the online processing of S_Profile, D_Group and S_Group methods is reasonably efficient. The average processing time per query is about **0.70** millisecond for S_Profile and S_Group, and **1.09** millisecond for D_Group. We aim to further improve the efficiency of the D_Group method by applying parallel programming in our future work.

[1] The expression $\sum_{d \in IN_{u,v}} p(t|d)$ in Eq. (9) is not dependent on the input query and so can be calculated offline.

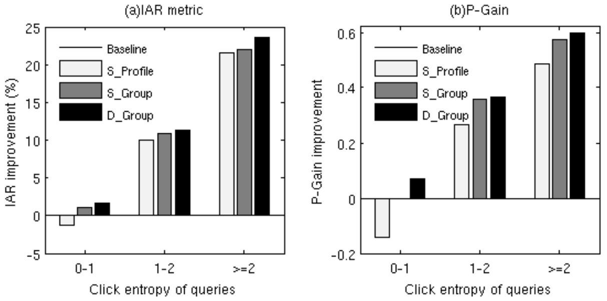

Figure 1: Search performance improvements over the baseline with different click entropies.

4.2 Performance on Different Click Entropies

In this section, we evaluate the search performance with respect to different click entropies (in Figure 1). Similar to [3], we find that when users have more agreements over clicked documents on these queries with small click entropy, and the personalisation methods even deteriorate the search performance. Specifically, with click entropies between 0 and 1, the IAR of S_Profile method is 1.31% lower than the baseline's. However, improvements are achieved by using the group-based methods (S_Group and D_Group). As seen in Figure 1, the improvement of personalised search performance increases significantly when the click entropy becomes larger, especially with click entropies ≥ 1. Furthermore, the personalisation methods achieve highest improvements when click entropies are no less than 2. In this case, all three personalisation methods have improvements of about 22% in IAR score over the baseline. These results indicate that the higher click entropy is, the better performance the search personalisation is likely to achieve. Moreover, in general D_Group performs better than S_Group, and both D_Group and S_Group methods outperform S_Profle's ($p \ll 0.01$).

4.3 Performance on Different Group Sizes

We also investigate the impact of the group size on the performance of the D_Group method. Table 4 shows the search performance using dynamic grouping method against different numbers of nearest neighbours. Due to the small size of dataset (106 users), forty-five (42.5%) users in the dataset only have common clicks (as used to calculate $p(u, v|q)$ in Eq. (9)) with no more than five other users. Therefore, we test the number of similar users from 1 to 5 in this experiment[2]. The results show that in general, the more users share common interests with the current user, the higher performance the D_Group tends to achieve.

Table 4: The performances of D_Group method over the different group sizes.

Group Size	IAR	P-Gain
0/S_Profile	*0.3679*	*0.1579*
1	0.3726	0.2613
2	0.3734	0.2651
3	0.3740	0.2930
4	0.3744	0.3130
5	**0.3755**	**0.3253**

This indicates that the information from user groups is useful. Even with only one other user in the group, the performance of the D_Group method achieves improvements of 1.28% (IAR) and 65.48% (P-Gain) over S_Profile where user profiles are not enriched by group information. With five nearest users, the D_Group method achieved the highest performance: improvements of 2.07% ($p \ll 0.01$) in IAR and 106% in P-Gain over S_Profile.

5. CONCLUSIONS AND FUTURE WORK

We have presented a framework for search personalisation using dynamic grouping method to enrich a user profile. For a user, the profile is dynamically constructed and enriched with information from other users whose interests are similar to the user given a query. Applying it to web search, we use the enriched profile to re-rank search results. Our experimental results demonstrate that the proposed method can stably and significantly improve the ranking quality. In the future, we plan to extend the model to capture user's interests that change over time. We will also carry out evaluation on larger-scale data sets.

6. ACKNOWLEDGEMENTS

We thank anonymous reviewers for their valuable comments. We would also like to thank Prof. Ji-Rong Wen and Dr. Zhicheng Dou for providing the experimental data. This work is funded in part by the Chinese National Program on Key Basic Research Project (973 Program, grant no. 2014CB744604) and the Natural Science Foundation of China (grant no. 61272265).

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 ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, pages 185–194, NY, USA, 2012. ACM.

[2] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003.

[3] 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 '07, pages 581–590, NY, USA, 2007. ACM.

[4] S. Fox, K. Karnawat, M. Mydland, S. Dumais, and T. White. Evaluating implicit measures to improve web search. *ACM Trans. Inf. Syst.*, 23(2):147–168, 2005.

[5] 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 Conference on Information and Knowledge Management*, CIKM '13, pages 2309–2314, NY, USA, 2013. ACM.

[6] A. K. McCallum. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu, 2002.

[7] K. Raman, P. N. Bennett, and K. Collins-Thompson. Toward whole-session relevance: Exploring intrinsic diversity in web search. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, pages 463–472, NY, USA, 2013. ACM.

[8] J. Teevan, M. R. Morris, and S. Bush. Discovering and using groups to improve personalized search. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, WSDM '09, pages 15–24, USA, 2009. ACM.

[9] R. W. White, W. Chu, A. Hassan, X. He, Y. Song, and H. Wang. Enhancing personalized search by mining and modeling task behavior. In *Proceedings of the 22nd International Conference on World Wide Web*, WWW '13, pages 1411–1420, Switzerland, 2013. ACM.

[2] We will test larger group sizes in our future work when carrying out further evaluation with larger-scale data sets.