Topic Model based Privacy Protection in Personalized Web Search

Wasi Uddin Ahmad, Md Masudur Rahman, Hongning Wang
Department of Computer Science, University of Virginia, VA USA
{wua4nw, mr5ba, hw5x}@virginia.edu

ABSTRACT

Modern search engines utilize users' search history for personalization, which provides more effective, useful and relevant search results. However, it also has the potential risk of revealing users' privacy by identifying their underlying intention from their logged search behaviors. To address this privacy issue, we proposed a *Topic-based Privacy Protection* solution on client side. In our solution, each user query will be submitted with k additional cover queries, which will act as a proxy to disguise users' intent from a search engine. The set of cover queries are generated in a controlled way so that each query carries similar uncertainty to randomize a user's search history while still providing necessary utility for the search engine to perform personalization. We used statistical topic models to infer topics from the original user query and generated cover queries of similar entropy but from unrelated topics. Extensive experiments are performed on AOL search log and the promising results demonstrated the effectiveness of our solution.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

Keywords

Privacy, Information Retrieval, Personalized Search

1. INTRODUCTION

Modern search engines, such as Google, Bing, Yahoo, exploit logged users' search behaviors to get insights of their search intents for personalization purpose. Although exploitation of search logs is helpful for improving search effectiveness, but the possibility of building individual user profile raises the concern of privacy breach. Anonymization of the search log data does not solve the problem. In 2006, AOL released an anonymized search query log of around 600,000 randomly selected users. The logs had been anonymized (at the server side) by removing individually identifying information such as IP address, username and any other personal information associated with that user. However, the actual queries and their corresponding query time, clicked URL and the anonymous ID were used to identify the gender, age and location of users [6]. Merely hiding a user's identity is not enough, but we need to hide a user's true search intent to ensure privacy. Obfuscate a user's true search intent to a search engine is very difficult: we need to first identify the search intent, properly embellish it before submitting to the search engine, such that the returned search results are still useful. As our preliminary attempt in this direction, we propose a *Topic-based Privacy Protection* (TPP) approach to enhance privacy in personalized web search. We have to admit that there are many different personalization techniques employed in commercial search engines; in this work, we assume personalization is achieved by server-side constructed user profiles [11].

Our proposed solution is client centered, and no facility is required on the search engine side. When a user submits a query to the search engine, we send k additional queries which act as a surrogate to randomize a user's profile. We refer to those additional queries as cover queries, since they cover or hide a user's true intent from the search engine. By adding noise through cover queries, we force search engine to have lower perception about the users. By varying the configuration of the generated cover queries, e.g., entropy, query length, topic proportion etc., we can affect the precision of the search engine constructed user profile as well as the level of personalization it can provide. The more cover queries are submitted with the original query, the less likely the user profile constructed on the search engine side will disclose a user's privacy (but it will be also less useful for personalization).

We use probabilistic topic models to infer users' search intent from their issued queries. In particular, we employed Latent Dirichlet Allocation (LDA) model [2] to infer topic proportion of the original query, and treat the inferred topics as a proxy of users' search intent. Then we create a set of cover queries by sampling queries from different topics. To ensure the plausibility of the automatically generated cover queries, we used entropy to measure the specificity of the cover queries and also varied their length with Poisson distribution. One advantage of our method is that the topic model can be trained on an isolated corpus, e.g., news archive, such that 1) the model can be readily deployed to new users without a requirement of pre-training; 2) the generated cover queries could be evenly distributed and sufficiently remote from a user's true intent.

2. RELATED WORKS

Complete privacy preservation is possible through *Private Information Retrieval* (PIR) [4], but its high complexity and inability to perform targeted personalized search prevent its practical adoption in commercial search engines. Server-controlled privacy is also assumed to protect user privacy; but after AOL search log release incident[1], several methods were proposed for improved query log anonymization. Researchers also proposed different user controlled or client-centered approach for privacy preservation. *Providing Privacy through Plausibly Deniable Search* (PDS) [7] is one of the client-centered privacy preserving approaches, and it is closely re-

SIGIR '16, July 17–21, 2016, Pisa, Italy.
© 2016 ACM. ISBN 978-1-4503-4069-4/16/07…$15.00
DOI: http://dx.doi.org/10.1145/2911451.2914753

lated to our proposed solution. In PDS, Latent Semantic Indexing is used to generate cover queries. PDS constructs a predefined set of all possible cover queries in an offline manner, while our method generates the cover queries on the fly. In addition, PDS does not consider generating standard varieties of cover queries in case of sequentially edited queries. One of the major bottleneck of PDS is that it cannot submit user query to the search engine if the query does not contain words in the predefined dictionary.

In [12, 3], better search results can be achieved with privacy guarantee if personalization is only performed based on a less sensitive or less specific part of the user profile, namely a generalized profile. The main idea is to build a hierarchical user profile and not to expose the sensitive part of the profile to the search engine by acquiring the level of privacy requirement from the user. [13] automatically builds a hierarchical user profile in the client side based on user specified privacy settings. In *Knowledge-based Scheme* [10] a similar approach is proposed to generate distorted user queries from a semantic point of view in order to preserve the utility of user profiles. In addition, linguistic analysis techniques are used to properly interpret complex queries submitted by users and generate new semantically-related queries accordingly. [8] proposed a different way to protect user privacy by embellishing the search queries with decoy terms that exhibit similar specificity spread as the genuine terms, but point to plausible alternative topics. [14] concentrated only on anonymizing user profiles by clustering them into user groups by taking into account the semantic relationships between query terms while satisfying the privacy constraints. Our proposed model is different from the aforementioned works as it concentrates on obfuscating users' true search intent at the topic level, which is constructed based on external data. And the balance between privacy protection and utility of search results is achieved at this topic-level obfuscation.

3. METHODOLOGY

Search engines track different type of user information, such as browsing history, clicked documents, amount of time spent in exploring a returned document, to build the user profile for personalizing the search results. Our proposed solution, *Topic-based Privacy Protection* (TPP), focuses on obfuscating the user profile by submitting cover queries along with the original query to the search engine. As a result, the user profiles constructed on the search engine side contain irrelevant information about the user such that it reduces the confidence of the search engine to predict users' specific information need (achieve privacy protection). By controlling noise injection from client side, TPP maintains the balance between privacy protection and search effectiveness. Figure 1 describes the work flow of our proposed solution. In our work, we inject k cover queries with the original query and submit them to the search engine. After getting the search results for all the submitted queries, we only keep the results of the original query and re-ranking them based on the user profile constructed and maintained on the client side. Finally, the re-ranked search results are provided to the user. In this way, we prevent the search engine to infer users' true information need with high fidelity.

3.1 Cover Query Generation

The amount of cover queries determines the strength of privacy protection. The more diverse those cover queries are, the more difficult it is for a search engine to distinguish individual user's information needs. But it will also result in degenerated search results. Therefore, balancing privacy preservation and search effectiveness is important. We generate cover queries based on the topics inferred by LDA topic model. We used a large collection of BBC news data set [5] for LDA model training and generate k cover queries randomly from the learned topics. Cover queries are generated on different topics with similar entropy to the original query's entropy e, e.g., $[e-\epsilon, e+\epsilon]$. Hence, if a user's query is highly concentrated in sports, less cover queries will be generated from the topic of sports but more will be generated from business, entertainment etc. If we failed to get a query within the required entropy range after n attempts, we will select the last generated cover query. To increase the plausibility of cover queries, we also randomize the length of cover queries by a Poisson distribution, where the rate parameter is set to the target user's average query length. We also generated random dummy clicks for the cover queries (with the same expectation of the number of clicks in true queries) so that search engines will not have explicit signal to recognize them. Therefore, our model disguises a user's true search intents through plausible cover queries such that search engines cannot easily recognize them.

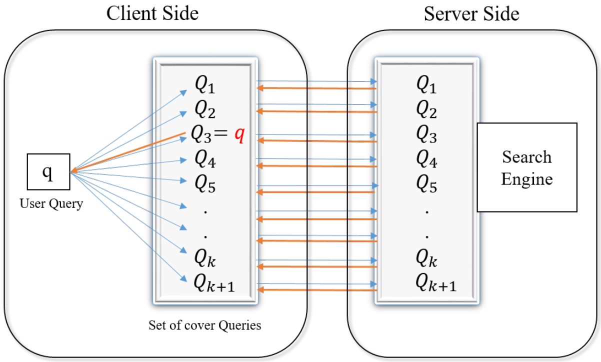

Figure 1: Workflow of Topic-based Privacy Protection solution.

3.2 Improving Search Effectiveness

To improve the utility of search results after cover query injection, we also build user profiles on client-side with a user's true queries and clicks for search result re-ranking. We assume a user's previous search queries and the corresponding clicked documents are good proxies of a user's search interests. Following the method proposed in [11], we use language models to build user profiles. In particular, we update the user profile immediately after each user query and result click. All the computation is performed at the client side, and no additional facility is required from search engine side. Once getting the search results, we only consider the results of the true user query, and re-rank the returned documents based on a linear combination of two scoring functions as shown in Eq (1).

$$\text{Score(d)} = \alpha \sum_{t} P(t) \times TF(t) + (1-\alpha)\frac{1}{\text{R}} \qquad (1)$$

The first part is based on the true user profile constructed and maintained by the client side, where $P(t)$ is the probability of observing term t in user profile and $TF(t)$ is the term frequency of t in document d. The second part is based on the ranking R of the documents provided by the search engine. Because search engine is forced to use an obfuscated query log for personalization, client-side re-ranking will help to improve the utility of resulted ranking, given the client-side user profile is built on the users' true query history and result clicks.

4. EXPERIMENTS

We performed extensive experiments using AOL search log released in 2006 [9]. We built our own search engine based on Apache Lucene and compared our model with two other previous works [7, 8, 10, 14] for performance evaluation. To the best of our knowledge, no previous work has validated their solutions in terms of both search effectiveness and privacy preservation.

4.1 Dataset & Setup

The AOL search log contains 20M search queries from 0.65M users from March to May 2006. There are 1,632,797 unique clicked URLs. We crawled all those URLs using an open source web crawler, *crawler4j*. We found approximately 64.4% (1,051,483) of the URLs are alive and the rest are no longer active. We only collected the text content of each URL to build the index of our search engine using *Apache Lucene*, where *Okapi BM25* is employed for ranking. Just for simplicity purpose, our search engine always returns the top 100 documents and thus we calculated mean average precision (MAP) at 100 to evaluate ranking quality. To personalize the search results, our search engine re-ranks them using the server-side constructed user profiles before returning the results. In particular, the server-side user profiles are also constructed using each submitted user query and the corresponding clicked document content (with cover queries and dummy clicks). We have selected the top t words using tf-idf weight from clicked documents content to update user profile. In evaluation, we considered the top 250 users based on the size of their query history. We only used the unique queries from each user, and all the corresponding clicked URLs are considered as relevant when measuring the search effectiveness. This gives us 45,200 queries over 250 users. The reason to remove duplicated queries in each user is that both our method and baselines will generate different cover queries for repeated queries with high probability, and this makes it easy to recognize those generated cover queries.

To build the topic model, which is the core of our cover query generation procedure, we used BBC dataset [5]. This data set contains news articles of five major topics, namely, business, entertainment, politics, sports and technology. There are 2,225 news articles and 23,225 unique terms in this dataset. We evaluated our model by varying the number of cover queries from 1 to 5. We also experimented with 3 different entropy range (ϵ = 0.1/0.2/0.3). To evaluate whether the cover queries are disclosing any information about a user's original query, we computed mutual information (MI) defined in Eq (2) between them. On the other hand, we also used Kullback–Leibler divergence as in Eq (3) between the true user profile on client-side and the noisy user profile on server-side to measure the amount of privacy disclosure.

$$MI(X;Y) = \sum_{y\epsilon Y}\sum_{x\epsilon X} p(x,y) log(\frac{p(x,y)}{p(x)p(y)}) \qquad (2)$$

$$D_{KL}(P\|Q) = \sum_{i} P(i) \log \frac{P(i)}{Q(i)} \qquad (3)$$

4.2 Results

We compared our model with Plausible Deniable Search (PDS) [7] and Knowledge-based Scheme (KBS) [10], and the proposed TPP outperformed both of them in terms of search effectiveness and privacy preservation. The detailed comparison between our model and the baselines is presented in Table 1 under different number of cover queries.

We implemented the PDS model on BBC dataset and evaluated it in our search engine. The total number of seed queries was 13,668 and total number of canonical queries was 13,340. The number of PD query sets found at level 0 and level 1 was 6735 and 3367 respectively. Since PDS creates cover queries through hierarchical clustering, only 2^n number of cover queries can be generated and thus we cannot report the performance of it for $k = 1, 3, 5$ cases in Table 1. The major bottleneck of PDS is that it fails to generate any cover query if a query term can not be found in the predefined dictionary (no query will be submitted to search engine then). Another limitation of PDS is that the PD query sets do not cover all words in the predefined dictionary, and oftentimes the user query and the closest canonical queries does not have any similarity in their content which results in very poor retrieval performance. We improved their solution by submitting the original user query along with the cover queries generated. As a result, we got improved MAP but smaller KL divergence and higher mutual information which assert that our model is more effective than PDS.

Table 1: Comparison between TPP, PDS and KBS

Settings	Model Name	MAP	KL Divergence	MI
K = 1	TPP	0.1389	0.0496	0.7761
	PDS	NA	NA	NA
	KBS	0.0331	2.2723	0.6972
K = 2	TPP	0.1388	0.0981	0.8523
	PDS	0.1386	0.2281	1.0195
	KBS	0.0361	2.2658	1.0869
K = 3	TPP	0.1388	0.1473	0.9681
	PDS	NA	NA	NA
	KBS	0.0363	2.3525	1.1591
K = 4	TPP	0.1387	0.1894	1.0339
	PDS	0.1386	0.2414	1.0114
	KBS	0.0368	2.3965	1.2541
K = 5	TPP	0.1386	0.2292	1.1413
	PDS	NA	NA	NA
	KBS	0.0364	2.3923	1.3237

KBS relies on structured knowledge modeled in the form of ontology, and it focuses on nouns and noun phrases when analyzing user queries. We implemented this method based on WordNet and ODP categories [10], which are organized in hierarchical structures. According to KBS, a new query set is constructed from a semantic point of view using predefined hierarchical structure of topical categories. Since the original query is not submitted to search engine in KBS, it can only provide limited search quality to users. In its original paper, KBS model is not validated for search effectiveness but we evaluated its performance through our implemented search engine. It is important to note that, KBS only uses the category name in the predefined hierarchy as cover queries and as a result, the resulting retrieval performance is extremely bad. The generated cover queries in KBS are more specific compared to those from our model (since it is already a summary of users' search intents), which might give the search engine reasonable amount of information about user intents.

TPP has two parameters: the number of cover queries k, and the entropy range ϵ. Though the impact of entropy range is not evident in the average MAP across users, it is evident in individual user's MAP. Increase in entropy range (e.g., from 0.2 to 0.3) decreases MAP around 1% for some users as cover queries become more diverse. We have also evaluated TPP with different number of topics (e.g., 5, 7, 9) during topic model training and got very similar results (in terms of MAP), which indicates the robustness of TPP with respect to the specific topic model used. We also tested TPP for k=0 to verify how much TPP is affecting the search effectiveness, and found that the decrease in MAP is negligible. Moreover, We tested TPP without client side re-ranking and surprisingly we got better MAP in that scenario: with client side re-ranking, we got a MAP of 0.123 while the MAP without the client side re-ranking is 0.138. Though for some users, client side re-ranking greatly improved the MAP but for many users it has fallen short to ensure better MAP. One major reason for this degenerated retrieval performance after client-side re-ranking is due to our pre-processing, where we have removed the duplicated queries from each user but such profile-based personalization mostly improves repeated queries. In our future work, we will evaluate our model with larger number of users with their full search log.

We also computed the difference between Information Content (IC) [10] of true user query and a corresponding cover query generated in TPP, PDS and KBS. The degree of IC between original and cover query is evaluated as the ratio between the query with the highest hit count with respect to the other. We picked 1500 user queries randomly and their corresponding two cover queries, in total of 3000 query pairs from each model to calculate IC ratio which is depicted in Figure 2. We used Microsoft Bing API to find the hit counts of queries. As shown in Figure 2, information content ratio of our model is smaller compared to PDS and KBS, because in PDS cover queries are generated from frequent patterns and in KBS hierarchical category names from ODP is used to generate cover queries. Since higher IC ratio means the specificity of original query and cover queries are not similar, the cover queries generated in PDS and KBS may reveal user privacy which is handled in TPP.

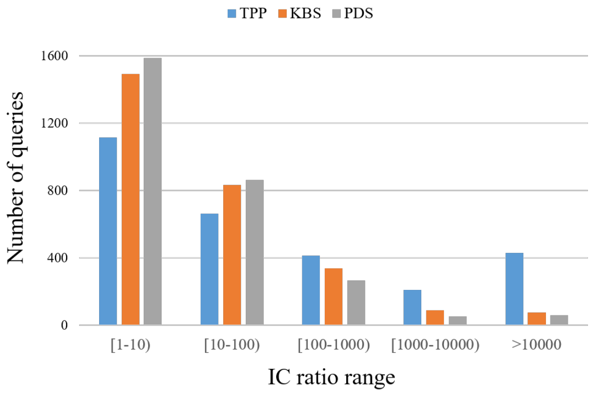

Figure 2: Information Content comparison between TPP, KBS and PDS

5. DISCUSSIONS

The number of generated cover queries are predefined in our model. But we should note that different users would need different level of privacy protection, and these trade-offs might also vary across different search tasks. We can rely on users to set their privacy level but it would be more useful if we can assess the level of protection required from a user's previous search history and behaviors. Moreover, we should be able to estimate the most feasible entropy range over a period of time to have more balanced personalization and privacy protection. We are learning the topic models only on news data set now, but increasing the diversity of training data sets, e.g., combining data from different sources (e.g., user reviews, social media, and forum discussions) would give us a more comprehensive topic model to generate cover queries.

6. CONCLUSIONS AND FUTURE WORKS

In this work, we developed a novel solution to protect user privacy based on their inferred search intent from topic models. The topic model, as a core component of our solution, can be estimated on an isolated document collection, which ensures protection of individual users' privacy and its general applicability. Both the specificity and length of the generated cover queries are carefully controlled to ensure the plausibility of cover queries. We experimented with 250 users over their 3 months' search history, in total of 45,200 queries from the AOL search log to prove the effectiveness of our model. The proposed method improved both search effectiveness and privacy protection against two state-of-the-art baselines.

Our current solution generates cover queries independently from the search context, e.g., queries in the same session and a user's previous clicks. This will inevitably hurt not only the plausibility of cover queries but also search effectiveness. As our future work, we will explicitly model a user's sequential search behaviors for generating better cover queries. In addition, we will also explore how to control the cover query generation dynamically such that the trade-off between personalization and privacy can be optimized for the long run. In addition, our current solution does not handle users' *ego-surfing* behaviors, such as searching for their own names or social security numbers. Classifiers can be built to recognize such queries and generate cover queries of the same type accordingly.

7. ACKNOWLEDGMENTS

We thank the anonymous reviewers for their insightful comments. This paper is based upon work supported in part by a Yahoo Academic Career Enhancement Award and the National Science Foundation under grants IIS-1553568.

8. REFERENCES

[1] M. Barbaro, T. Zeller, and S. Hansell. A face is exposed for aol searcher no. 4417749. *New York Times*, 9(2008):8For, 2006.

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

[3] G. Chen, H. Bai, L. Shou, K. Chen, and Y. Gao. Ups: efficient privacy protection in personalized web search. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 615–624. ACM, 2011.

[4] B. Chor, E. Kushilevitz, O. Goldreich, and M. Sudan. Private information retrieval. *Journal of the ACM (JACM)*, 45(6):965–981, 1998.

[5] D. Greene and P. Cunningham. Practical solutions to the problem of diagonal dominance in kernel document clustering. In *Proc. 23rd International Conference on Machine learning (ICML'06)*, pages 377–384. ACM Press, 2006.

[6] D. Halse et al. I know what you did last summer. *Advocate: Newsletter of the National Tertiary Education Union*, 21(1):14, 2014.

[7] M. Murugesan and C. Clifton. Providing privacy through plausibly deniable search. In *SDM*, pages 768–779. SIAM, 2009.

[8] H. Pang, X. Ding, and X. Xiao. Embellishing text search queries to protect user privacy. *Proceedings of the VLDB Endowment*, 3(1-2):598–607, 2010.

[9] G. Pass, A. Chowdhury, and C. Torgeson. A picture of search. In *InfoScale*, volume 152, page 1, 2006.

[10] D. Sánchez, J. Castellà-Roca, and A. Viejo. Knowledge-based scheme to create privacy-preserving but semantically-related queries for web search engines. *Information Sciences*, 218:17–30, 2013.

[11] X. Shen, B. Tan, and C. Zhai. Implicit user modeling for personalized search. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 824–831. ACM, 2005.

[12] L. Shou, H. Bai, K. Chen, and G. Chen. Supporting privacy protection in personalized web search. *Knowledge and Data Engineering, IEEE Transactions on*, 26(2):453–467, 2014.

[13] Y. Xu, K. Wang, B. Zhang, and Z. Chen. Privacy-enhancing personalized web search. In *Proceedings of the 16th international conference on World Wide Web*, pages 591–600. ACM, 2007.

[14] Y. Zhu, L. Xiong, and C. Verdery. Anonymizing user profiles for personalized web search. In *Proceedings of the 19th international conference on World wide web*, pages 1225–1226. ACM, 2010.