Recommending Diverse and Relevant Queries with A Manifold Ranking Based Approach

Xiaofei Zhu[1,2,3] Jiafeng Guo[1] Xueqi Cheng[1]

[1]Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China
[2]Graduate School of the Chinese Academy of Sciences, Beijing 100049, China
[3]School of Mathematics and Statistics, Chongqing University of Technology, Chongqing 400054, China
{zhuxiaofei, guojiafeng}@software.ict.ac.cn, cxq@ict.ac.cn

ABSTRACT

Query recommendation has been considered as an effective way to help search users in their information seeking activities. Traditional approaches mainly focused on recommending alternative queries with close search intent to the original query. However, to only take the relevance into account may generate redundant recommendations which provide almost the same information for users. Therefore, it is important to provide diverse as well as relevant query recommendations. In this way, we are able to cover multiple potential search intents of users and attract more clicks over recommendations. Besides, previous query recommendation approaches mostly relied on measuring the relevance or similarity between queries in the Euclidean space. However, there is no convincing evidence that the query space is Euclidean. Therefore, it is more natural and reasonable to assume that the query space is a manifold. In this paper, we aim to recommend diverse and relevant queries based on the intrinsic query manifold. We propose a unified model, named manifold ranking with stop points, for query recommendation. Specifically, by introducing stop points into query manifold, our approach can iteratively rank queries for recommendation by simultaneously considering both diversity and relevance between queries in an unified way. Empirical experimental results show that our approach can effectively generate highly diverse as well as closely related query recommendations.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Algorithm, Experimentation

Keywords

Query Recommendation, Manifold Ranking, Click-through Data

1. INTRODUCTION

With the exponential growth of information on the Web, search engine has become an indispensable tool for Web users to seek their desired information. However, it is never easy for users to formulate a proper query to search because query is usually very short [4] and words are ambiguous [11]. Furthermore, users sometimes cannot express their search intent precisely due to the lack of domain-specific knowledge. Therefore, how to help users formulate a suitable query has been recognized as a challenging problem. To overcome this problem, a valuable technique, query recommendation, has been employed by most commercial search engines, such as *Google, Yahoo!*, and *Bing* to improve usability.

Traditional query recommendation approaches mainly focused on recommending alternative queries with close search intent to the original query. Query logs are widely used in these approaches [2, 7, 11], where similar queries are identified based on users' historical behavior and used as recommendations for each other. However, to only take the relevance/similarity into account may generate redundant recommendations. For example, when a user issues a query 'abc', the system may recommend him/her 'abc television' and 'abc tv', which are both very relevant to 'abc' but of the equivalent meaning. Recommending such queries at the same time will decrease the recommendation quality since they provide almost the same information to users. Therefore, it is important to provide diverse as well as relevant query recommendations. By reducing the redundancy, we are able to cover multiple potential search intents of users and thus attract more clicks over recommendations.

In addition, previous query recommendation approaches mostly relied on measuring the similarity between queries in the Euclidean space, either based on query terms or click-through data. However, there is no convincing evidence that the query space is Euclidean. Inspired by the research work on document modeling [12, 13], it is more natural and reasonable to assume that queries are sampled from a nonlinear low-dimensional manifold which is embedded in the high-dimensional ambient space. The local geometric structure is essential to reveal the relationship between queries.

In this paper, we aim to recommend diverse and relevant queries based on the intrinsic query manifold. We propose a novel unified model, named manifold ranking with stop points, for query recommendation. Specifically, our approach leverages a manifold ranking process over query manifold in essentials, which can naturally make full use of the relationships among queries to find relevant and salient queries. More important, we introduce the stop points into query manifold to capture the diversity during the ranking process. Therefore, our approach can generate query recommendations iteratively by simultaneously considering both diversity and relevance between queries in a unified way. Empirical experimental results show that our approach can effectively generate highly diverse as well as closely related query recommendations.

2. RELATED WORK

Query recommendation. Query recommendation has been em-

SIGIR Workshop on Query Representation and Understanding '10, July 23, 2010, Geneva, Switzerland

ployed as a core utility by many industrial search engines. Most of the work on query recommendation is focused on measures of query similarity, where query log data has been widely used in these approaches. Beeferman et al. [2] applied agglomerative clustering algorithm to the click-through bipartite graph to identify related queries for recommendation. Wen et al. [11] proposed to combine both user click-through data and query content information to determine the query similarity. Li et al. [7] recommended related queries by computing the similarity between queries based on query-URL vector model and leveraging a hierarchical agglomerative clustering method to rank similar queries.

However, most previous work only focused on recommendation relevance, while not explicitly addressed the problem of diversity. Mei et al. [8] tackled this problem using a hitting time approach based on the Query-URL bipartite graph. Their approach can recommend more diverse queries by boosting long tail queries. However, the weakness of their approach is that it would sacrifice the relevance considerably when improving the diversity, and many long tail queries recommended to users may not be familiar to them.

Manifold ranking. Manifold ranking was used to rank data with respect to the intrinsic global manifold structure collectively revealed by a huge amount of data [12, 13]. It has been applied in many research fields [6, 10, 9] recently where a ranking is needed in essentials. For example, He et al. [6] leveraged manifold ranking to measure relevance between the query and database images for image retrieval. Wan et al. [10] applied the manifold ranking process to utilize the relationships between the topic and the sentences for text summarization. However, so far there is no related work on applying manifold ranking for query recommendation.

3. OUR APPROACH

3.1 Notation

Given a set of data points (i.e. queries) $\mathcal{X} = \{q_0, q_1, \ldots, q_n\} \subset \mathbb{R}^m$, the first point q_0 is the input query and the rest of the points q_i ($1 \leq i \leq n$) are the candidate queries. Hereafter, query and point will not be discriminated unless otherwise specified. Let $d : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ denote a metric on $\mathcal{X}$ (e.g. Euclidean distance), where $d(q_i, q_j)$ is the distance between q_i and q_j. Let $f : \mathcal{X} \rightarrow \mathbb{R}$ denote a ranking function which assigns to each point q_i ($0 \leq i \leq n$) a ranking value f_i. We can view f as a vector $f = [f_0, \ldots, f_n]^T$. We also define a vector $y = [y_0, \ldots, y_n]^T$, in which $y_0 = 1$ for the input query q_0 and $y_i = 0$ ($1 \leq i \leq n$) for all the candidate queries.

3.2 Query Manifold

In our work, queries are assumed to be sampled from a nonlinear low-dimensional manifold which is embedded in the high-dimensional ambient space. To build up such a query manifold, we first need to identify the *K Nearest Neighbors* for each query. Here we leverage the click-through information in query logs for this purpose.

The click-through data can help us find similar queries. The basic idea is that if two queries share many clicked URLs, they have similar search intent to each other [7]. Therefore, we model queries in terms of query-URL vectors, instead of query-term vectors. We represent each query q_i as a L_2-normalized vector, where each dimension corresponds to one unique URL in the click-through data. Specifically, given a query q_i ($0 \leq i \leq n$), the j-th element of the feature vector of q_i is

$$q_i^j = \begin{cases} \frac{e_{ij}}{\sqrt{\sum_{k=1}^{M} e_{ik}^2}} & \text{if } q_i \text{ clicked } u_j; \\ 0 & \text{otherwise,} \end{cases}$$

where M denotes the total number of unique URLs in the click-through data and e_{ij} denotes the weight for the pair of query q_i and its clicked URL u_j. Here we follow the CF-IQF weighting scheme [5] and define the weight $e_{ij} = cf_{ij} \times \log(N/qf_j)$, where cf_{ij} denotes the total click frequency on u_j given q_i, qf_j denotes the total number of unique queries which have clicked u_j, and N denotes the total number of unique queries in the query log. The distance between two queries q_i and q_j is then measured by the Euclidean distance between their normalized feature vectors

$$d(q_i, q_j) = \sqrt{\sum_{k=1}^{M} (q_i^k - q_j^k)^2}.$$

With the definitions above, we construct the query manifold as follows. Firstly, each query is represented as a data point on the manifold. We then connect any two points with an edge if they are among the K nearest neighbors to each other ($K = 50$ in our case). In this way, we are able to preserve the sparse property of the query manifold. We define an affinity matrix W for the query manifold, where $w_{ij} = exp[-d(q_i, q_j)^2/2\sigma^2]$ if there is an edge linking q_i and q_j, and $w_{ii} = 0$ as there are no loops in the graph. Here σ is empirically set to 1.25.

3.3 Manifold Ranking with Stop Points

A traditional manifold ranking process over the query manifold can be described as follows:

1. Symmetrically normalize W by $S = D^{-1/2} W D^{-1/2}$ in which D is the diagonal matrix with (i, i)-element equal to the sum of the i-th row of W.
2. Iterate $f^{(t+1)} = \alpha S f^{(t)} + (1 - \alpha) y$ until convergence, where α is a parameter in $[0, 1)$.
3. Let f_i^* denote the limit of the sequence of $\{f_i^{(t)}\}$. Rank each point q_i according its ranking scores f_i^* (largest ranked first).

In the above ranking process, all the points spread their ranking scores to their neighbors via the weighted graph. The spread process is repeated until a global stable state is achieved, and all the points are ranked according to their final ranking scores. With the traditional manifold ranking process, we can obtain relevant and salient queries for recommendation given the input query.

To explicitly address the diversity of query recommendation, we introduce stop points into query manifold and propose a novel ranking approach named manifold ranking with stop points. The stop points are a special type of points on query manifold, which stop spreading their ranking scores to their neighbors during the manifold ranking process. Intuitively, we can imagine the stop points as the "black holes" on the manifold, where no ranking scores would be able to "escape" from them. By turning queries already selected for the recommendation into stop points, the ranking scores of other queries close to these queries will be naturally penalized during the ranking process based on the intrinsic query manifold.

Here we derive the new iteration algorithm for manifold ranking with stop points. Let T denote the set of stop points, and R denote the set of free points (all data points excluding the stop points). The normalized matrix S in traditional manifold ranking can then be reorganized as a block matrix $\begin{pmatrix} S_{RR} & S_{RT} \\ S_{TR} & S_{TT} \end{pmatrix}$, and the original iteration equation in step 2 can be written as

$$\begin{pmatrix} f_R \\ f_T \end{pmatrix}^{(t+1)} = \alpha \begin{pmatrix} S_{RR} & S_{RT} \\ S_{TR} & S_{TT} \end{pmatrix} \begin{pmatrix} f_R \\ f_T \end{pmatrix}^{(t)} + (1 - \alpha) \begin{pmatrix} y_R \\ y_T \end{pmatrix},$$

where f_R and f_T denotes the ranking scores of points in set R and T respectively, and y_R and y_T denotes the prior on the points in set R and T respectively.

Since stop points never spread their scores to their nearby points, we set $S_{RT} = S_{TT} = 0$, then we get the new iteration equation for

manifold ranking with stop points:

$$\begin{pmatrix} f_R \\ f_T \end{pmatrix}^{(t+1)} = \alpha \begin{pmatrix} S_{RR} & 0 \\ S_{TR} & 0 \end{pmatrix} \begin{pmatrix} f_R \\ f_T \end{pmatrix}^{(t)} + (1-\alpha) \begin{pmatrix} y_R \\ y_T \end{pmatrix}.$$

As we turn the queries already selected for recommendation into stop points, the ranking scores of stop points are no longer useful for us since the stop points would not be selected later again. All we care about is the ranking scores of the free points in set R. Therefore, we only need to compute f_R with the iteration equation

$$f_R^{(t+1)} = \alpha S_{RR} f_R^{(t)} + (1-\alpha) y_R. \quad (1)$$

THEOREM 1. *The sequence* $\{f_R^{(t)}\}$ *converges to*

$$f_R^* = (1-\alpha)(I - \alpha S_{RR})^{-1} y_R.$$

PROOF. Without loss of generality, suppose $f_R^{(0)} = y_R$. By iteration equation (1), we have

$$f_R^{(t)} = (\alpha S_{RR})^t y_R + (1-\alpha) \sum_{i=0}^{t-1} (\alpha S_{RR})^i y_R.$$

Let $P = D_{RR}^{-1} W_{RR}$, P is the similarity transformation of S_{RR} as follows:

$$S_{RR} = D_{RR}^{-1/2} W_{RR} D_{RR}^{-1/2} = D_{RR}^{1/2} D_{RR}^{-1} W_{RR} D_{RR}^{-1/2} = D_{RR}^{1/2} P D_{RR}^{-1/2}$$

hence P and S_{RR} have the same eigenvalues. Let λ be an eigenvalue of P, according to the Gershgorin circle theorem, we have

$$|\lambda - P_{ii}| \leq \sum_{j=1, j\neq i}^{|R|} |P_{ij}|,$$

where $|R|$ is the size of the free point set. Note that $P_{ii} = 0$ and $\sum_{j=1,j\neq i}^{|R|} |P_{ij}| \leq 1$, so we have $|\lambda| \leq 1$. Since $0 \leq \alpha < 1$ and $|\lambda| \leq 1$, then $\lim_{t\to\infty} (\alpha S_{RR})^t = 0$, $\lim_{t\to\infty} \sum_{i=0}^{t-1} (\alpha S_{RR})^i = (I - \alpha S_{RR})^{-1}$. Hence, we have

$$f_R^* = \lim_{t\to\infty} f_R^{(t)} = (1-\alpha)(I - \alpha S_{RR})^{-1} y_R.$$

□

3.4 Recommendation Approach

Based on the ranking algorithm above, we finally obtain our query recommendation approach. Give an input query, we build up a query manifold and set all the query points as free points. We then apply manifold ranking with stop points over the query manifold until a global stable state is achieved, and rank the queries according to their ranking scores. The free point with the largest ranking score (except the input query) will be selected as a recommendation, and set as a stop point in the following iteration. The process iterates until a pre-specified number of recommendations acquired. The recommendation algorithm using manifold ranking with stop points is shown in Algorithm 1. Note that although f_R^* can be expressed in a closed form, for large scale problems, the iteration algorithm is preferable due to computational efficiency.

Algorithm 1 Query Recommendation using Manifold Ranking with Stop Points

Input:
q - the input query
χ - all the other queries
K - recommendation size
S - normalized affinity matrix of the query manifold
T - stop point set
R - free point set
Output: Top K recommendation query set U
Initialization: $U = \phi, T = \phi, R = \chi$

1: **for** $k = 1 \ldots K$ **do**
2: obtain S_{RR} based on S, T and R.
3: iterate $f_R^{(t+1)} = \alpha S_{RR} f_R^{(t)} + (1-\alpha) y_R$ until convergence started with $f_R^{(0)} = 0$, where α is a parameter in [0,1).
4: select the query q_k with the largest ranking score (except the input query) as a recommendation, $U = U \cup \{q_k\}$.
5: turn query q_k from free point into stop point, $T = T \cup \{q_k\}$ and $R = R - \{q_k\}$.;l
6: **end for**

4. EXPERIMENTS

4.1 Data Set and Baselines

Our experiments are based on the Microsoft 2006 RFP dataset[1] which contains about 15 million queries (from US users) that were sampled over one month in May, 2006. We cleaned the raw data by ignoring non-English queries, converting letters into lower case, and trimming of each query. To further reduce the noise in clicks, the click-through between a query and a URL with a frequency less than 3 was removed. After cleaning, we obtained the click-through data with totally 191,585 queries, 251,427 URLs and 318,947 edges. On average, each query clicks 1.66 distinct URLs, and each URL is clicked by 1.27 distinct queries. As [3], we randomly sampled 150 queries with frequencies between 700 and 15,000 for evaluation.

We leverage three other query recommendation approaches as baselines: (1) Naive approach (*Naive*), which recommends the most relevant queries by measuring the similarity between queries in the Euclidean space. (2) Manifold ranking approach(*Mani_only*), which directly recommends top ranked queries by applying a manifold ranking process over query manifold. (3) Hitting-time approach (*Hitting_time*), which recommends queries using hitting time based on the Query-URL bipartite graph [8]. We refer our approach as *Mani_stop*, and the parameter α was fixed at 0.99 as used in [12, 13] in our experiments.

[1]http://research.microsoft.com/users/nickcr/wscd09/

4.2 Evaluation Metrics

Evaluating the quality of query recommendation is difficult, since there is usually no ground truth of recommendations and different annotators will have different judgements over the recommendation results. Therefore, we propose an automatic evaluation over query recommendation for more objective comparison between different approaches. Specifically, we leverage the Open Directory Project (ODP) and a commercial search engine (i.e., Google) to help evaluate the relevance and diversity of recommendations respectively.

Relevance. We adopt the same method used in [1] to evaluate the relevance of recommended queries. Specifically, we measure the relevance of two queries based on the similarity between their corresponding categories provided by ODP. Given two queries q and q', let C and C' denote the corresponding set of top k ($k = 10$ in our case) ODP categories from Google Directory. We define the similarity between two categories $c \in C$ and $c' \in C'$ as the length of their longest common prefix $l(c, c')$ divided by the length of the longest category of c and c'. More concisely, denoting the length of a category c with $|c|$, the similarity between two categories c and c' is $Sim(c, c') = |l(c, c')|/max\{|c|, |c'|\}$. The relevance between query q and q' is then defined as $r(q, q') = \max_{c\in C, c'\in C'} Sim(c, c')$. For an input query q, the relevance of its recommendations is defined as $rel(q) = \frac{1}{|U|} \sum_{q'\in U} r(q, q')$, where U denotes the recommendation set and $|U|$ is the number of queries in U.

Diversity. We measure the diversity of recommended queries based on the differences between their top ranked search results provided by Google. Specifically, given two queries q and q', we compute the proportion of different URLs among their top k ($k = 10$ in our case) search results by $d(q, q') = 1 - |o(q, q')|/k$, where $o(q, q')$ is the number of overlapped URLs among the top k search results of query q and q'. Then for an input query q, the diversity of its recommendations is defined as $div(q) = \sqrt{\frac{\sum_{q\in U} \sum_{q'\in U} d(q,q')}{|U|(|U|-1)}}$.

Table 1: Recommendation for an example query 'abc' by using four different approaches.

Naive	Hitting_time	Mani_only	Mani_stop
'abc shows'	'abc shows'	'abc tv'	'abc tv'
'abc television'	'abc television'	'abc news'	'abc news'
'abc tv'	'associated builders and contractors'	'abc family'	'abc nightline'
'abc news'	'abc tv'	'abc shows'	'abc family'
'abc breaking news'	'news stories'	'abc breaking news'	'associated builders and contractors'

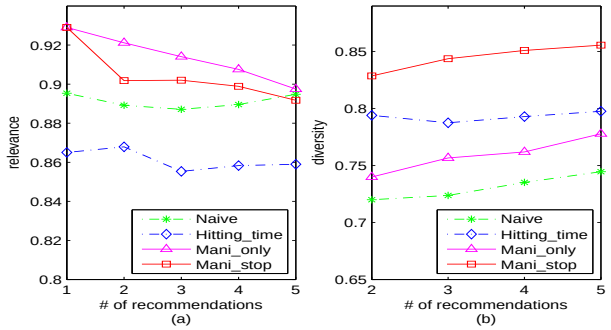

Figure 1: (a) Average relevance of query recommendation over different recommendation size under four approaches. (b) Average diversity of query recommendation over different recommendation size under four approaches.

4.3 Results and Discussion

We first show the top 5 recommendations for a sampled query 'abc' in Table 1 to demonstrate the differences between our approach and baselines. From Table 1, we observe that Naive approach recommended closely related but somewhat redundant queries, e.g. 'abc television' and 'abc tv', or 'abc news' and 'abc breaking news'. Hitting_time and Mani_only recommended queries with better diversity although there is still some redundancy , e.g. 'abc television' and 'abc tv' in Hitting_time, and 'abc news' and 'abc breaking news' in the Mani_only. Besides, the recommendation 'news stories' provided by Hitting_time seems not so closely related to the original query 'abc'. Finally, we can see that the Mani_stop recommended more diverse as well as closely related queries.

The automatic evaluations were conducted over a variety of recommendation size (up to top 5) and the results are presented in Figure 1. Figure 1(a) shows the average relevance of recommendations under the four different approaches. We note that both Mani_only and Mani_stop can achieve better or comparable performance as compared with Naive approach (i.e., using global similarity in Euclidean space). It demonstrates the effectiveness of the intrinsic query manifold in capturing the relevance between queries. However, the relevance of Hitting_time is consistently lower than that of all the other three approaches at each recommendation size. We conducted the T-Test ($p = 0.05$) over the results of Hitting_time and found that the performance drop is significant as compared with the other three. It shows that by boosting long tail queries for query recommendation, the relevance of recommendations would be considerably hurt.

The diversity of query recommendations of the four different approaches is shown in Figure 1(b)[2]. Not surprisingly, the diversity of Naive is the lowest one in the four approaches, since Naive only focuses on recommending queries according to their similarity with the input query. Mani_only gets better diversity than Naive. The major reason is that Mani_only tends to assign relevant and salient queries higher ranking scores. To boost queries with structure salience may implicitly bring in some diversity in recommendation. Hitting_time leverages the hitting time from candidate queries to the input query as their ranking scores, and thus boosts the long tail queries for recommendation and increases the diversity. Therefore, it obtains a higher diversity than both Naive and Mani_only. However, the disadvantage of Hitting_time is that it sacrifices the relevance considerably (See Figure 1(a)) when improving the diversity. Among all these four approaches, Mani_stop obtains the highest diversity as it explicitly address the diversity problem by introducing the stop points into query manifold. We also conducted the T-Test ($p = 0.05$) over the results of Mani_stop and found there is a significant improvement in diversity as compared with all the other three approaches.

The results in Figure 1 clearly demonstrate that our approach Mani_stop can effectively generate highly diverse as well as closely related query recommendations.

[2]The diversity at one recommendation is not shown here due to that it is meaningless with the diversity metric.

5. CONCLUSIONS

We have proposed a novel manifold ranking based approach to recommend diverse and relevant queries. Unlike previous methods, our approach leverage the intrinsic global manifold structure to measure the similarity of queries. Moreover, we introduce the stop points into query manifold, and thus simultaneously capture the diversity and relevance of query recommendation in a unified way. Experimental results show that our approach can recommend more diverse queries than baseline methods, while maintain a high relevance. For the future work, it may be interesting to explore different ways of constructing the query manifold and investigate how it may affect the recommendation performance.

6. REFERENCES

[1] R. Baeza-Yates and A. Tiberi. Extracting semantic relations from query logs. In *KDD '07*, pages 76–85, 2007.

[2] D. Beeferman and A. Berger. Agglomerative clustering of a search engine query log. In *KDD '00*, pages 407–416, 2000.

[3] P. Boldi, F. Bonchi, C. Castillo, D. Donato, and S. Vigna. Query suggestions using query-flow graphs. In *WSCD '09*, pages 56–63, 2009.

[4] H. Cui, J.-R. Wen, J.-Y. Nie, and W.-Y. Ma. Probabilistic query expansion using query logs. In *WWW '02*, pages 325–332, 2002.

[5] H. Deng, I. King, and M. R. Lyu. Entropy-biased models for query representation on the click graph. In *SIGIR '09*, pages 339–346, 2009.

[6] J. He, M. Li, H.-J. Zhang, H. Tong, and C. Zhang. Manifold-ranking based image retrieval. In *MULTIMEDIA '04*, pages 9–16, 2004.

[7] L. Li, Z. Yang, L. Liu, and M. Kitsuregawa. Query-url bipartite based approach to personalized query recommendation. In *AAAI'08*, pages 1189–1194, 2008.

[8] Q. Mei, D. Zhou, and K. Church. Query suggestion using hitting time. In *CIKM '08*, pages 469–478, 2008.

[9] R. Ohbuchi and T. Shimizu. Ranking on semantic manifold for shape-based 3d model retrieval. In *MIR' 08*, pages 411–418, 2008.

[10] X. Wan, J. Yang, and J. Xiao. Manifold-ranking based topic-focused multi-document summarization. In *IJCAI'07*, pages 2903–2908, 2007.

[11] J.-R. Wen, J.-Y. Nie, and H.-J. Zhang. Clustering user queries of a search engine. In *WWW '01*, pages 162–168, 2001.

[12] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In *NIPS' 03*, 2003.

[13] D. Zhou, J. Weston, A. Gretton, O. Bousquet, and B. Schölkopf. Ranking on data manifolds. In *NIPS' 03*, 2003.