Encoding Group Interests With Persistent Homology for Personalized Search

Yuchen Meng, Rong-Hua Li, Hongchao Qin, Xiang Wu, Huanzhong Duan, Yanxiong Lu, and Guoren Wang

***Abstract*—Personalized search aims to customize search results based on users' search history. The key of personalized search is to learn the representations of users' interests from users' search history. The state-of-the-art personalized search methods often encode group-level features of similar users to improve personalized search. However, existing group-level feature encoding methods are sensitive to noisy users, which are often contained in real-world search data. To overcome this problem, we propose a novel approach to encode group features based on a topological data analysis technique, namely, persistent homology analysis. Such topological features are typically robust to noisy data, thus can improve the personalized search quality. To the best of our knowledge, we are the first to use topological features for improving personalized Web search. We conduct extensive experiments on two real-life datasets to evaluate the proposed approach; and the results show that our solution is significantly better than the state-of-the-art personalized search models in terms of several widely used precision measures.**

***Index Terms*—Persistent homology, personalized search, rips complex, rips filtration.**

I. Introduction

SEARCH engine is a useful tool using in our daily life to get specific information from the Web. Given a document list and a query, the main task of a search engine is to return a ranked document list which is ordered by the relevance with the query. However, returning the same result to all users is obviously not the optimal choice, because with the same query, different users may have different query intents. That is why we need personalized search, which aims to adjust the ranking list for each user by the search engine. By returning a specific ranking list to each user based on his/her interests, the result can better fit the user's intent, thus improving the personalized search quality.

Previous studies [3], [12], [15], [17], [21], [43] usually build users' profiles based on their historical behaviors and take the relevance between the profiles and documents as the ranking score. Some of the other existing methods are based on groups with similar users and use these group-level profiles to enrich the user's profile [36], [37], [42], [47]. Most of these group-based methods generate the group-level profiles based on the topic similarities between queries and documents.

Such group-based methods have several limitations. First, although the users' profiles are extracted based on their historical behaviors, both the interest and the knowledge base of the users often evolve over time. That is to say, two users have similar knowledge level before, but they may have significantly different knowledge level now. Therefore, it is often very hard to represent a user's interest or knowledge base when he/she has similar behaviors with another user from his/her full history. Second, the topic similarity between the queries and documents used in existing group-based methods [36], [37], [42], [47] may involve too many noisy users. For example, consider a similar user group which is constructed based on the same hot-news documents clicked by the users [14]. Then, there may exist many users clicked those hot documents even when they did not have similar interests in this area; they were just clicking to take a look and they can probably never click documents in this area again. Clearly, such a similar user group may contain too many noisy users, thus reducing the performance of the group-based personalized search methods.

To alleviate the limitations of existing group-based methods, we propose a novel approach to construct user groups and derive the group profile features based on a topological data analysis technique, namely, persistent homology analysis. Such persistent-homology-based topological features are often robust to the noisy data, thus can significantly improve the performance of personalized search. To summarize, the main contributions of this article are as follows.

1) Instead of using the entire users' click history, we propose to use snapshots to extract the users' profiles, where a snapshot denotes a fixed-length partial click history of a user, and each user's click history can be divided into a set of consecutive snapshots. Compared to existing methods that uses the entire click history, the advantage of our snapshot representation is that it can better reflect the interests of users at a certain time, because users' interests are often evolve over time.
2) To construct a more reliable and helpful user group, we propose to use topological features, extracted by a persistent homology analysis algorithm, to calculate the similarity between snapshots. Armed with the constructed groups, we can derive the users' group profiles

Manuscript received 8 February 2024; accepted 25 May 2024. Date of publication 26 June 2024; date of current version 20 August 2024. This work was supported in part by the National Key Research and Development Program of China under Grant 2021YFB3301301, and in part by NSFC under Grant U2241211 and Grant 62072034. This article was recommended by Associate Editor N. T. Nguyen. *(Corresponding author: Rong-Hua Li.)*

Yuchen Meng, Rong-Hua Li, Hongchao Qin, Xiang Wu, and Guoren Wang are with the Beijing Institute of Technology, Beijing 100081, China (e-mail: meng_yc@163.com; lironghuabit@126.com; qhc.neu@gmail.com; 3120211060@bit.edu.cn; wanggrbit@gmail.com).

Huanzhong Duan and Yanxiong Lu are with the Wechat Search Application Department, Tencent, Shenzhen 518054, China (e-mail: boosterduan@tencent.com; alanlu@tencent.com).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TSMC.2024.3410029.

Digital Object Identifier 10.1109/TSMC.2024.3410029

and then use the group features to improve personalized search. To the best of our knowledge, our work is the first to use the topological features for improving personalized search.

3) We conduct comprehensive experiments to evaluate our method using two large-scale real-world query log datasets. The results demonstrate that our solution significantly outperforms the state-of-the-art personalized search model. For example, on the AOL dataset, our method can improve the search precision by 7.6% over the state-of-the-art model.

II. Related Work

A. Personalized Search

The goal in personalized search is ranking the documents to fit the user's interest and his current aim which were extracted from the user's search logs. The key problem is how to build users' interest profile from their search history. The existing research generally focuses on the following two directions.

Traditional Personalized Search Models: Early studies on personalized search focus mainly on constructing the features from search logs, such as click features. Dou et al. [16] predicted the click probability of a certain document by counting the times the documents are clicked in the user's search history. Some other models [8], [20], [33], [41] try to extract the features in a topic space which is constructed from the clicked documents and queries. Another line of the studies [5], [43] focus on using the current query and users' history to extract features and put these features into a ranking model. However, these methods generally perform poorly compared to machine learning approaches.

Deep Learning-Based Personalized Search: Except for the traditional methods, the deep-learning-based personalized search methods are also used for personalized search which can be roughly classified into two different types. The first type of method is the adaptation framework [35], the other kind of method is to learn an explicit representation of the user interest profile from the click history. For example, Lu et al. [24] devised PSGAN which take the generative adversarial network (GAN) into the personalized searching model. Yao et al. [46] proposed PEPS which applies personal word embedding for personalized search. However, most of the methods use only the history and current query, the knowledge level is seldom considered in personalized features.

B. Group-Based Personalized Search

In addition to the methods using users' own search logs discussed above, group-based search aims to enhance the personalized search method with the similar search logs of other users. To extract similar search logs, there are two main existing methods, based on users' search behavior or based on users' social relations.

The search behavior-based methods correlate users according to their search history then enhance the search method. The G-Click model proposed by Dou et al. [16] can find users with the most similar search behavior and using these users to score the documents. Morris et al. [26] tried using group behaviors to enhance the quality of traditional Web search. Teevan et al. [36] tried different grouping ways and find out that grouping method could help identify the documents that users thought more relevant to the query. Vu et al. [41] proposed to construct the group dynamically in response to the input query. This model considered the users' different interests when facing different topics and build the corresponding group.

The second method were proposed to correlate users based on the social relations and using these relation to enhance the search method. Bender et al. [25] designed an approach to exploit social relations. Specifically, they put the users, tags, and documents into a friendship graph and combining semantic and social signals then applied PageRank on it. Similarly, Kashyap et al. [2] constructed six social groups and a social aware search graph for ranking. Bjorklund et al. [38] and Carmel et al. [13] build the social network with the help of social applications and using these social relations while ranking.

Some of the studies tried to combine these two method to get a better result. Zhou et al. [47] build two relationship network according to the users' search log and their social relationship extracted from the social chat app, respectively. These models achieve the state-of-the-art performance by using the personal and group interest profile. Unlike these work, we propose a novel topological-based approach to enhance users' interest profiles so as to improve personalized search performance. However, for the group-based method, the search history of noisy users can usually affect the model significantly.

C. Persistent Homology

Previous works have tried to link the different users to enhance the performance of ranking method. However, these previous group constructions are mostly not stable or need additional data support. For example, grouping users based a generally clicked documents can cause the users with less similarity gathered in the same group. As a result, many of the results are easily been affected by the noisy data or cannot be used on the data without social relation information. Our proposed method, by contrast, applies tools from statistical persistent homology theory to build stable user groups using only the search logs of users. Persistent homology is a topological data analysis technique which has been successfully applied to medical image analysis and brain network analysis [10], [11], [22], [23], [28], [34]. It can extract topological features from a network or a point cloud data, captures structural information from the data using the language of algebraic topology. The extracted information is a kind of powerful and stable feature because it weakens the characteristics of specific document in the users' logs. Such feature can be used in various contexts, such as medical field [9], [27], sensor network coverage [40], and cosmology [44]. Recently, it has also been used for image segmentation [45], clustering and graph representation learning [19]. For graph-structured data, topological features have been used for node classification and graph classification. In this work, we extend the applications of the topological data analysis method to extract topological features in personalized search. To the best of our knowledge, we are the first to use the topological data analysis technique for personalized search.

III. Proposed Method

As discussed in the Introduction, most existing models can only represent users' interests while ignoring their knowledge base, rendering that the search engine may misunderstood users' search intents. For example, if a man is searching a kind of cosmetic, he is more likely trying to figure out what it is. However, if he is a makeup artist, he may try to figure out how to make up with it better. To handle this problem, we propose a new model, called persistent homology-based personalized search (PHPS), to find users who not only have the same interest categories, but also have similar knowledge level in a specific interest category. These categories can be extracted based on the documents clicked by the users. Specifically, for each category, we first partition the user's search history (including a set of query-document pairs) into a set of fixed-length partial search history (e.g., 20 query-document pairs as a partial search history). We refer to such a fixed-length partial search history as a snapshot. Then, we construct a group of users' snapshots in different categories, and then we aggregate the most similar snapshots in the same category which we called *friend snapshots* in the following. Based on these *friend snapshots*, we can build a group-level profile for a user. The detailed architecture of our model is shown in Fig. 1, which will be interpreted in the following sections.

A. User Personal Interest Profile Construction

We make use of a user's click history in the current search session and user's full click history, respectively, to construct the user's interest profile. Here, the search session means a short time period from the user begin to use the searching function until he stop using it for a fixed time. These two parts of click history are corresponding to the short-term user profile and long-term user profile, respectively.

Let $H_u^l = \{(q_1, d_1), \ldots, (q_n, d_n)\}$ be the full click history of a user u, which is also referred to as the long-term history of u. Each item in H_u^l is a pair consisting of a query and a clicked document. We also consider the recent click history of a user as his/her short-term history, denoted by $H_u^s = \{(q_{k+1}, d_{k+1}), \ldots, (q_n, d_n)\}$. Here, k is the split point of the click history. In our method, we define the beginning of the under-going search session as the split point.

To construct user personal profile, we use a pretrained sentence transformer [29] to turn the query text and the clicked document into vectors, which are denoted by q and d, respectively. Such sentence transformer is a kind of BERT network using Siamese and triplet networks. It can tackle the query ambiguity and document noise at word level. Moreover, it can combine contextual information from the neighboring words to enhance the sentences vectors. It shows the state-of-art performance on sentence classification and sentence-pair regression tasks, so we choose it as the structure to convert the query text and clicked documents. With these vectors, we use $s_i = q_i \oplus d_i$ to represent a click behavior of a user, where $\oplus$ denotes the concatenate operator.

With the representation of the click behavior, we are capable of representing the short-term history as $H_u^s = \{s_{k+1}, s_{k+2}, \ldots, s_n\}$ and take it as the input of the short-term profile $\mathsf{Transformer}$

$$p_{\text{short}} = \mathsf{Transformer}_{short}\left(H_u^s, P\left(H_u^s\right)\right) \tag{1}$$

where $P(\cdot)$ denotes the position embedding of the vectors in H_u^s. $\mathsf{Transformer}_{short}$ is a $\mathsf{Transformer}$ encoder whose last output is the user's short-term profile represented as p_{short}.

Similarly, we can get the long-term history H_u^l as the input of the long-term profile $\mathsf{Transformer}$

$$p_{\text{long}} = \mathsf{Transformer}_{long}\left(H_u^l, P\left(H_u^l\right)\right) \tag{2}$$

where the $\mathsf{Transformer}_{long}$ is a $\mathsf{Transformer}$ encoder and p_{long} is the last output of the $\mathsf{Transformer}$ representing the user's long-term profile. $P(\cdot)$ is the position embedding of the vectors s_i.

B. User Group Interest Profile Construction

For a user u, we use the snapshots of his/her short-term click history to identify the most similar snapshots of the other users in the same category. For convenience, we refer to such similar snapshots as the *friend snapshots* for u. All the documents contained in the *friend snapshots* of u will be used to extract the group interest profile for the user u. The key point is how to define a similar metric to identify the *friend snapshots*. A simple solution is to use a cosine similarity metric to measure the similarity between the representations of two snapshots. Such a method, however, is often sensitive to noisy users, because some hot documents in the snapshots may be clicked by too many *noisy users* in real-world search scenarios (as analyzed in the Introduction). In this section, we propose a novel persistent-homology-based method to define a similarity metric to measure the similarity between two snapshots. Such a persistent-homology-based method is often robust with respective to noisy data.

Document Graph Construction: To perform persistent homology analysis, we first construct a document graph from the search history of all users, which models the adjacency relationships between the documents in users' search history. Specifically, we build a graph $\mathcal{G}_i$ for each category c_i which contains all clicked documents in the category c_i. Each node in $\mathcal{G}_i$ represents a clicked document in the category c_i. If two nodes (i.e., documents) in $\mathcal{G}_i$ are clicked by the same user, we add an edge between these two nodes. For each edge (d_i, d_j), we compute the similarity weight of (d_i, d_j), denoted by $R(d_i, d_j)$ as follows. First, we initialize $R(d_i, d_j) = 0$. Then, if the document d_i and d_j are clicked by the same user, we add an quality r_{user} to $R(d_i, d_j)$. Further, if both d_i and d_j are clicked by a user in the same search and in the same session, we add r_{search} and $r_{session}$ to $R(d_i, d_j)$, respectively. Here, r_{user}, r_{search} and $r_{session}$ are hyperparameters which satisfy $r_{user} \ll r_{search} = r_{session}$ (e.g., $r_{user} = 1$, $r_{search} = r_{session} = 1000$). This is because two documents appears in the same search or the same session should be much more similar than the two documents

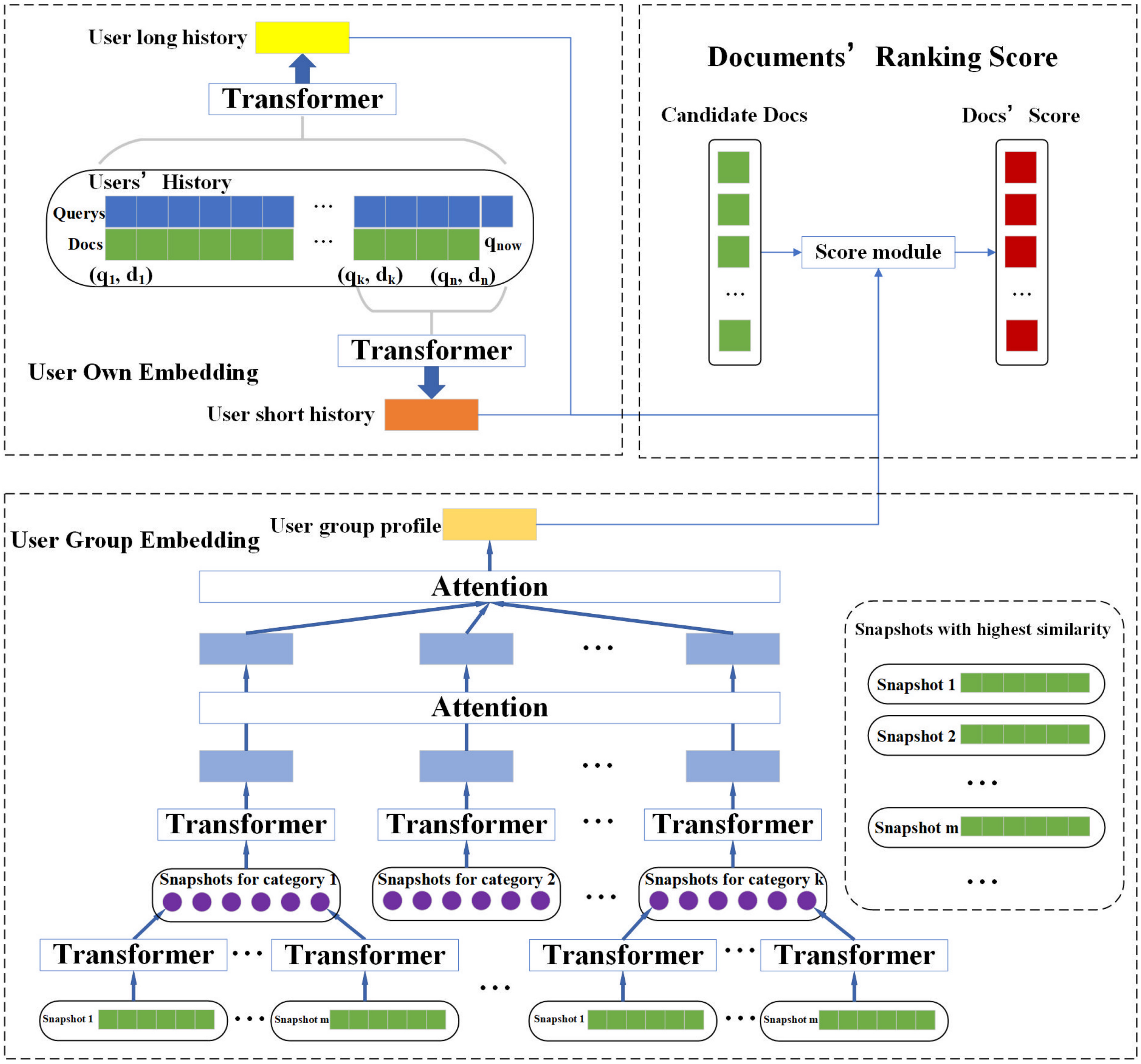


Fig. 1. Architecture of PHPS. We use users' click history, snapshots, and the current query as the inputs of the model. The model can be divide into three modules: 1) a module generates user's long-term profile and short-term profile from user's click history; 2) a module generates the user's group profile from the snapshots which have the highest similarity with the user's own snapshots; and 3) a module evaluates the scores of the candidate documents based on the extracted profiles.

clicked by the same users but within different search or sessions. For each edge (d_i, d_j), we define the distance between (d_i, d_j) as $D(d_i, d_j) = \mu/R(d_i, d_j)$, where μ is a positive constant. Algorithm 1 shows the detailed implementation of computing the distance between two adjacent documents.

Algorithm 1 $D(d_i, d_j)$ Between Two Adjacent Documents

```
Initialize $R(d_i, d_j) = 0, \mu > 0$
for $c_x, c_y$ in click history do
    if $c_x.userID = c_y.userID$ then
        $R(d_i, d_j) \leftarrow R(d_i, d_j) + r_{user}$
    end if
    if $c_x.sessionID = c_y.sessionID$ then
        $R(d_i, d_j) \leftarrow R(d_i, d_j) + r_{session}$
    end if
    if $c_x.searchID = c_y.searchID$ then
        $R(d_i, d_j) \leftarrow R(d_i, d_j) + r_{search}$
    end if
end for
$D(d_i, d_j) = \mu/R(d_i, d_j)$
```

Distance Computation for Any Pair of Documents: Note that by Algorithm 1, if $R(d_i, d_j) = 0$, we have $D(d_i, d_j) = +\infty$. That is, two nonadjacent documents in the graph $\mathcal{G}_i$ have infinite distance. In real-world search scenarios, however, some nonadjacent documents may also be highly similar. To remedy the limitation, we propose to compute the shortest-path distance between two documents in the graph $\mathcal{G}_i$. However, computing all-pair shortest path distances is often intractable for large graphs. To improve the efficiency, we propose a landmark-based heuristic method to approximate the shortest path distance. Specifically, we choose n_{lm} nodes in $\mathcal{G}_i$ as the landmarks, which are represented by $\mathcal{L} = \{l_1, l_2, \ldots, l_n\}$. Then, we use the Dijkstra algorithm on these landmark nodes to compute the single-source distance between the landmarks and all other nodes in $\mathcal{G}_i$. Using these landmarks and the shortest path distances, we can approximate the distance between any two nodes d_i, d_j which have $R(d_i, d_j) = 0$ by

$$D\big(d_i, d_j\big) = \min_{k=1,2,\ldots,n_{lm}} \big(D(l_k, d_i) + D\big(l_k, d_j\big)\big). \tag{3}$$

Simplicial Complex on Point Cloud: Before involve the persistent homology analysis into our model, we first introduce the basic concepts used in persistent homology and relate them to group interest profile constructing.

For a set of document point cloud data X with p points, we can compute the distance $D(d_i, d_j)$ between two points from the document graph. Let ϵ be a distance threshold. We give a definition of the network constructed by thresholding correlations between the nodes. If the two points d_i and d_j satisfies that $D(d_i, d_j) \leq \epsilon$, then we connect two points with an edge. The collection of all those edges is denoted as E. The graph consisting of the node set X and the edge set E is called binary network $\mathcal{B}(X, \epsilon)$ at threshold ϵ. This network can also be seen as a topology.

Generally, given a point cloud data set with a rule for connections, the topological space is a simplicial complex and its element is a simplex. To give a brief understanding, a node is a 0-simplex, an edge is a 1-simplex, and a triangle (including the space between the edges) is a 2-simplex. More generally, a complete graph with p nodes represents the edges of a $(p-1)$-simplex. Based on the description above, a simplicial complex is defined as following [23].

Definition 1: A simplicial complex $\mathcal{K}$ is a finite collection of simplices such that: 1) any face of $\sigma \in \mathcal{K}$ is also in $\mathcal{K}$ and 2) for $\sigma_1, \sigma_2 \in \mathcal{K}$, $\sigma_1 \cap \sigma_1$ is a face of both σ_1 and σ_2.

From the definition of simplicial complex, the binary network can also be seen as a simplicial complex consisting of 0-simplices (nodes) and 1-simplices (edges). There are many kinds of different simplicial complexes, in this article, we choose to employ the Rips complex which is defined as following:

Definition 2: Given a point cloud data X, the Rips complex $\mathcal{R}(X, \epsilon)$ is a simplicial complex whose k-simplices correspond to unordered $(k+1)$-tuples of points which are pairwise within ϵ distance.

The difference between binary network $\mathcal{B}(X, \epsilon)$ and Rips complex $\mathcal{R}(X, \epsilon)$ is that there can have at most 1-simplices (edges) in $\mathcal{B}(X, \epsilon)$ while there can have at most $(p-1)$-simplices in $\mathcal{R}(X, \epsilon)$. Such difference is shown in Fig. 2.

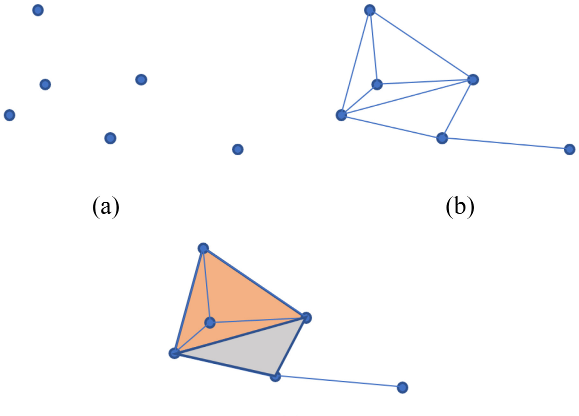


Fig. 2. Difference between point cloud, binary network, and Rips complex. (a) Set of point cloud data X with 6 points. (b) Binary network $\mathcal{B}(X, \epsilon)$ based on X at threshold ϵ. (c) Rips complex $\mathcal{B}(X, \epsilon)$ based on X at threshold ϵ. The orange part represents a 3-simplex and the gray part represents a 2-simplex, which are the simplices not shown in the $\mathcal{B}(X, \epsilon)$.

Persistent Homology Analysis: Based on the above approximate distances and the basic concepts of topology analysis, we can perform persistent homology analysis over the clicked documents. For each snapshot in a specific category c, we can regard all the documents in this snapshot as a point cloud s^c, where each document corresponds to a point. Let ϵ be a distance threshold. If the two points d_i and d_j satisfies that $D(d_i, d_j) < \epsilon$, then we connect two points with an edge.

As illustrated in Fig. 3(a), we can clearly see that when changing the threshold ϵ from zero to ∞, the graph derived from the point cloud data will change. Specifically, in Fig. 3(a), we begin with the distance threshold is 0. At this point, none of the nodes are connected with edge. When the distance threshold increase to 1, which is the distance between node x_1 and x_2, these two nodes are connected with an edge. Similarly, when it increase to 2, which is the distance between node (x_2, x_3) and (x_4, x_5), (x_2, x_3), and (x_4, x_5) are connected. The threshold ϵ keeps growing and while the ϵ grows to ∞, the connectivity between the points in the graph changes. We can finally obtain a sequence of graphs during such procedure which represented by the so-called Rips complex (a well-known concept in topological data analysis [11])

$$\mathcal{R}\left(s^c, \epsilon_0\right) \subseteq \mathcal{R}\left(s^c, \epsilon_1\right) \subseteq \cdots \subseteq \mathcal{R}\left(s^c, \epsilon_n\right) \tag{4}$$

where $\epsilon_0 < \epsilon_1 < \cdots < \epsilon_n$. $\mathcal{R}(s^c, \epsilon_i)$ denotes a Rips complex based on the point cloud data s^c and the distance threshold ϵ. This sequence of Rips complex is called a *Rips filtration* which is the key component of persistent homology analysis [11], [34].

The topological changes of the Rips filtration can be represented by a set of *barcode* [7], [31], which is constructed by plotting the changing topological features over different filtration values. We choose the Betti number as the topological feature described by the barcode. By definition, the kth Betti number β_k shows the number of $(k+1)$-dimension holes in the graph. For example, zeroth Betti number β_0 shows the number of connected components in the graph, first Betti number β_1 shows the number of holes in the graph. For example, if we need the filtration that contains information of the nodes and edges, then we can use β_0 (information of components) to derive such a filtration.

Each bar represents the birth and death time of a specific topological feature. Fig. 3(c) illustrates a *barcode* generated by the *Rips filtration*. The length of the barcode shows the persistence of a specific topological feature. For example, the sixth barcode in Fig. 3(c) shows that one of the connected components in the graph birth at the beginning, and dead when ϵ grows to 1. That is because when the threshold ϵ grows to 1, node x_1 and x_2 are connected by an edge so the number of connected components went from 6 to 5. We can also visualize this evolving procedure by another way as illustrated in Fig. 3(d) which is called persistent diagram. It transforms the barcode into a point in the diagram which can also be used to derive the topological feature. Specifically, in Fig. 3(d), the blue node (1, 0) correspond to the sixth barcode in Fig. 3(c). The persistence of a specific topological feature reflects, to some extent, the importance of that feature across the entire

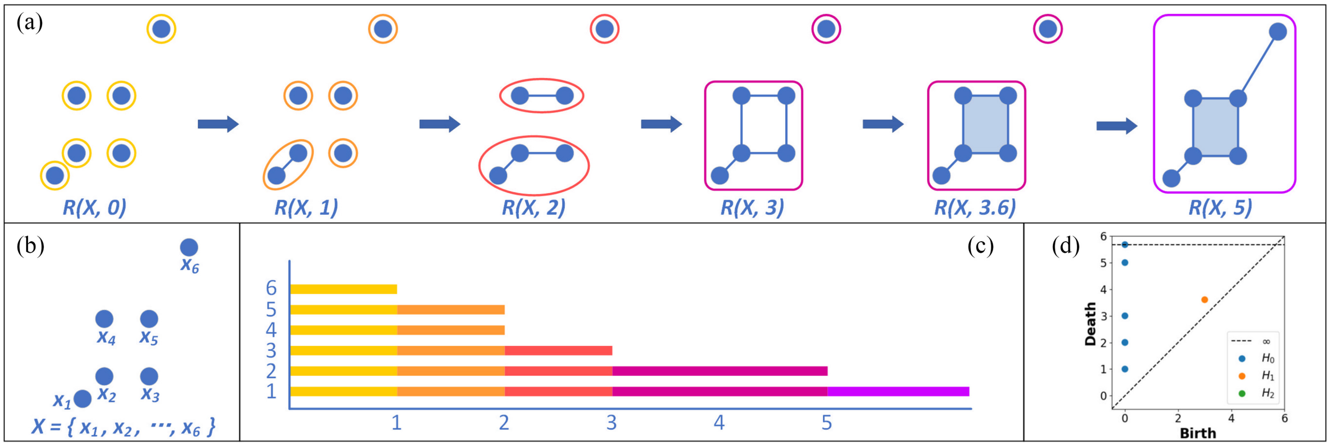


Fig. 3. Results of persistent homology analysis with barcode and persistent diagram. (a) Rips filtration generated from the point cloud data [shown in (b)] at different ϵ values (0, 1, 2, 3, and 5). (b) Point cloud data X. (c) Topological changes visualized by the barcode which represent the zeroth Betti number β_0. The horizontal axis represents the filtration value. Each of the barcode represents a connected component. (d) Topological changes visualized by the persistent diagram which represent the zeroth and first Betti number β_0, β_1. The H_1 point shown on the diagram represents the presence of a hole that born when the $x_2, x_3, x_4,$ and x_5 are connected as a rectangle and die when the $(x_2, x_5), (x_3,$ and $x_4)$ are connected.

graph because it implies that the feature will persist regardless of how significant the perturbations are. Therefore, we aim to use the persistently existing topological features on the graph as a basis for assessing the correlation between graphs.

After deriving the filtrations from the snapshots, we can define a similarity metric to measure the topological similarity between two snapshots based on the filtrations of these snapshots. Specifically, for two filtrations F, G in the same category, we can compute the distance between them by using the persistence scale-space kernel [30] which is defined as

$$k_\sigma(F, G) = \frac{1}{8\pi\sigma} \sum_{y \in F, z \in G} \exp\left(-\frac{\|y - z\|^2}{8\sigma}\right) - \exp\left(-\frac{\|y - \bar{z}\|^2}{8\sigma}\right) \tag{5}$$

where $k(\cdot)$ represents the scale-space kernel and $y \in F, z \in G$ are the pair of birth and death time. The notion $\bar{z}$ is defined as $\bar{z} = (b, a)$ if $z = (a, b)$, and σ is a hyperparameter. We choose this kernel mainly because it is defined via an L2-valued feature map which can maintaining the stability property of persistent homology. The scale parameter of the kernel also provides our method with a higher level of generality. Based on (5), we define the normalized similarity S as follows:

$$S_\sigma(F, G) = \frac{2k_\sigma(F, G)}{k_\sigma(F, F)k_\sigma(G, G)}. \tag{6}$$

Constructing Users' Group Interest Profile: With (5), we can compute the similarity between two snapshots by using a topological method. The next step is to use these snapshots to construct the users' group interest profile.

Specifically, for a user u, we can extract the most recent snapshot s^c of u in a certain category c. Then, we make use of the scale-space kernel to identify the n_f snapshots which has the highest similarity with s^c. We refer to these highly similar snapshots as *friend snapshots* of s^c. Based on those *friend snapshots*, we build a group interest profile for u as follows.

First, we use a Transformer [39] to generate a representation for each friend snapshot s^i as follows:

$$s_i = \text{Transformer}_{\text{snapshots}}(d_1, d_2, \ldots, d_{\text{m}}) \tag{7}$$

where $d_1, d_2, \ldots, d_m$ are the documents in the friend snapshot s^i. Equipped with the representations of these snapshots, we can use another Attention [39] layer to construct the user's group interest profile in the category c by

$$S^c = \text{Attention}_{\text{category}}(s_1, s_2, \ldots, s_{n_f}) \tag{8}$$

where s_i is a vector of the top n_f similar snapshots in c.

C. Personalized Search Model

Armed with all the users' profiles obtained by the proposed method, we are ready to present our personalized search model. First, for each user u, we build the user's group interest profile from his/her search history, denoted by $\Psi_u = \{S_u^{c_1}, S_u^{c_2}, \ldots, S_u^{c_n}\}$. Here, n is the number of the categories of the documents in u's search history. When the user u issues a query q, we put Ψ_u and the short-term profile together into a attention layer to get a new snapshot profile $p^{i,s}$ which captures both his/her group interest and his/her current search intent by the following way:

$$p^{i,s} = \text{Attention}_{i,s}(\Psi_u, p_{\text{short}}). \tag{9}$$

Second, we use the user's personal profiles and the group profile to calculate the similarity score with the document d. To obtain a more comprehensive representation of the document, we also use the additional features $\mathcal{F}_d$ extracted by the method proposed in [5]. We put these features directly into the scoring module, and obtain the final score from a MLP layer

$$\begin{aligned} \text{score}_i = \text{MLP}\big(&sim(p^{i,s}, d_i), sim(p_{\text{long}}, d_i), \\ &sim(p_{\text{short}}, d_i), \mathcal{F}_d\big) \end{aligned} \tag{10}$$

where the $sim(\cdot)$ denotes the cosine similarity metric. After obtaining the score for each document, we adopt the LambdaRank algorithm [6] for ranking in our personalized

TABLE I
SUMMARY OF DATASETS

Dataset	AOL Dataset	Commercial Dataset
Users	24,227	101,852
Queries	181,257	12,003,667
Sessions	90,345	8,162,282
Documents	224,448	27,982,299
Clicks	188,007	24,479,377

search model. The input of this model is a pair of documents in the candidate documents. Formally, each input pair includes a positive document d_i and a negative document d_j and their distance is computed with a sigmoid function denoted by x_{ij}. The loss function of the model is defined as the cross entropy between the real distance and the predicted one

$$\mathcal{L} = -|\Delta|\left(\overline{x_{ij}} log\left(x_{ij}\right) + \overline{x_{ji}} log\left(x_{ji}\right)\right) \tag{11}$$

where $\overline{x_{ij}}$ denotes the real distance and Δ represents the change of ranking quality when we change the document d_i and d_j. By minimizing the loss using the Adam optimizer, we can obtain our final personalized search model.

IV. Experiments

A. Experimental Setup

Datasets: We use two different search logs datasets in our experiments. The detailed statistics of the datasets are outlined in Table I. The AOL dataset is a public query log dataset which contains users' anonymous ID, search ID, the timestamp when the document is clicked, the URL that has been clicked and the query text. Following [1], [4], we divide the users' history into sessions according to the similarity between two consecutive queries and get session ID for the query log. To train our model, we also need irrelevant documents which can be obtained by a method used in [1] and [4]. By using this method [1], [4], we can obtain five documents, including both irrelevant and positive documents for each query in the train stage and 50 documents in the test stage. The original ranking result of this dataset can be derived by using the BM25 algorithm [32]. The dataset is divided into training, validation and testing sets with a ratio of 6:1:1.

We also collect a commercial dataset from WeChat to evaluate our model. WeChat is the biggest social network platform in China which also has a built-in search engine. We select 100 000 users from WeChat which contains a query log with more than 100 clicked docs in a time period from 4 September 2021 to 4 October 2021. For each user u, the search history of u contains the following information: an anonymous user ID, a query text, and a timestamp when the document is clicked, the rank of the documents returned by the search engine, a search ID, a session ID, and a boolean variable denoting whether the document is clicked or not, the primary category of the document. Differ from AOL, the session here is determined based on a user's consecutive search behavior within a certain period of time. Specifically, we define a session as the continuous search history of a user within a 30-min timeframe. The dataset is divided into training, validation and testing sets with a ratio of 4:1:1.

Baselines: In addition to the original ranking method (the BM25 algorithm in the AOL dataset and the WeChat original ranking algorithm in the commercial dataset), we also select several state-of-the-art methods as baselines.

1) *P-click [16]:* This method ranks the documents base on the number of clicks by the user under the same query, and generates the personalized results by fusing the original ranking.
2) *SLTB [5]:* It first extracts 102 features from the click history, including click-based features, topic-based features, short and long-term features, time decay, etc. They sent all these features to the ranking model and use the LamdaMart algorithm [6] to generate the ranking list.
3) *RPMN [49]:* This is a deep learning-based method uses two refinding parts: a) query-based refinding and b) document-based refinding to enhance user refinding behavior for personalized search.
4) *HTPS [48]:* It no longer generate the users' profile, instead, it first applies the click history to build a context, and then uses this context to disambiguate the queries of the users by a transformer encoder.
5) *PSGAN [24]:* It is a model based on a GAN framework to construct the embedding of queries which can well capture users' current search intents. Moreover, it can select the document pair more valuable for generating users' profile.
6) *HRNN [18]:* This is a model which dynamically builds short-term, long-term user interests and highlight relevant interests with a hierarchical RNN model and a query-aware attention layer. It is also the first model leverage sequential information with a deep learning framework.
7) *PSSL [50]:* It is a model that adopts a contrastive sampling method to extract self-supervised information from sequences of users' history. It also introduced a Pretrained model in the reranking stage.

In addition, it is worth mentioning that there also exists a group-based personalized search model [47]. This model, however, requires the users' friendship information to construct user groups, which is prohibited by WeChat due to the privacy policy. Since the AOL dataset also does not contain the friendship information between the users, we cannot compare our method with such a group-based personalized search model [47].

Evaluation Metrics: We use the commonly used ranking metrics mean average precision (MAP), mean reciprocal rank (MRR) and P@1 (precision@1) to evaluate different models. For ablation experiments, we also make use of the average rank, NDCG@3, NDCG@5, and NDCG@10 to evaluate the effect of different parts of our model.

Model Settings: In our model, we use a pretrained sentence-transformer [29] to get the sentence embedding of the document title and the query text. Specifically, we use a 384-dimension embedding for the AOL dataset and a 100-dimension embedding for the commercial dataset. Since we

TABLE II
COMPARISON OF PHPS AND THE BASELINES. THE PERCENTAGE REFLECTS THE IMPROVEMENTS OVER THE SOTA METHOD (PSSL)

Model	AOL Dataset						Commercial Dataset					
	MAP		MRR		P@1		MAP		MRR		P@1	
Ori.	.2504	-66.0%	.2596	-64.2%	.2596	-59.6%	.4045	-7.2%	.4470	-4.7%	.2674	-4.0%
P-Click	.4224	-42.4%	.4298	-40.8%	.3788	-41.1%	.4063	-6.8%	.4476	-4.6%	.2680	-3.7%
SLTB	.5072	-31.1%	.5194	-28.4%	.4657	-27.6%	.4176	-4.2%	.4579	-2.4%	.2707	-2.8%
HRNN	.5423	-26.3%	.5545	-23.6%	.4854	-24.5%	.4207	-3.5%	.4601	-1.9%	.2732	-1.9%
PSGAN	.5480	-25.5%	.5601	-22.9%	.4892	-24.0%	.4245	-2.6%	.4622	-1.4%	.2746	-1.3%
RPMN	.5926	-19.5%	.6409	-11.7%	.5322	-17.3%	.4288	-1.6%	.4623	-1.4%	.2743	-1.5%
HTPS	.7091	-3.6%	.7251	-0.1%	.6268	-2.5%	.4293	-1.5%	.4643	-1.1%	.2749	-1.1%
PSSL	.7359	-	.7258	-	.6431	-	.4358	-	.4690	-	.2784	-
PHPS	**.7479**	+1.6%	**.7535**	+3.8%	**.6921**	+7.6%	**.4507**	+3.4%	**.4804**	+2.4%	**.2862**	+2.8%

cannot obtain enough user history if we divide the documents in the AOL dataset into categories, we regard all documents as the same category. On the contrary, on the commercial dataset, we divide the documents into 43 categories which is defined by WeChat based on the documents' content. For all multihead attention layer, we use eight heads with 64-dimension for each head.

To generate the group profiles, we only use the zeroth Betti number to extract topological features. The reason why we only use the zeroth Betti number is that we hardly find hole or voids in the document graph in our datasets. In other words, most of the snapshots' first Betti number and second Betti number are 0 in our datasets. To compute the scale-space kernel, we set σ as 0.5. For the $\mathsf{Transformer}_{short}$ we use, we define it as a 2-layer encoder, where each layer consists of an 8-head multihead attention layer, and the hidden feature dimension is 512. In the commercial dataset, we build the snapshots by a sequence of 20 clicked documents and refresh the user's snapshots when he have clicked ten more documents. In the AOL dataset, since the length of users' history are much shorter than the commercial dataset, we build the snapshots by a sequence of ten clicked documents and refresh the user's snapshots when he have clicked ten more documents. During the training process, we set the batch size and the learning rate as a dynamic number to ensure that we have the same group profile in the same batch. We train the model for 10–20 epochs and choose the model weight which show the best result on the validation set, and choose the it is result on testing set as the final result.

B. Experimental Results

1) Overall Performance: Table II reports the results of different models. From Table II, we can obtain the following observations.

1) PHPS significantly outperforms all the baselines in terms of all metrics, which demonstrate the high effectiveness of our solution. For example, compared to the state-of-the-art PSSL model, our model improves the ranking quality by 3.8% and 2.4% in terms of the MRR metric, on the AOL dataset and the commercial dataset, respectively.
2) Compared to all the baselines, PHPS achieve much better performance in terms of the P@1 metric on the AOL dataset. The reasons could be that the users' clicks in the AOL dataset is highly centered. Most of the users' clicks in this dataset concentrate on a few URLs. Such click features are helpful for constructing the document similarity graph as used in our model, thus may boost the performance of our group-based personalized search model. Moreover, our method shows comparative performance in retrieval time compared to the PSSL.

2) Ablation Experiments: Our PHPS model has three main components: 1) the short-term profile generator; 2) the long-term profile generator; and 3) the group profile generator. Thus, we conduct a ablation experiment to study the effect of the short-term profile, long-term profile, and group profile features. The results on the AOL dataset and the commercial dataset are shown in Tables III and IV, respectively. As can be seen, without the group profile features, the performance of our model decreases significantly on both two datasets. For example, on AOL, the MAP and NDCG@3 decreases by 6.9% and 7.8%, respectively. We also observe that both the short-term and long-term profiles are useful for our model. In addition, we also observe that the effect of the short-term profile feature is not very significant to our model on both two datasets. The reason could be that the short-term profiles are partially included in the group profiles in our model, thus resulting in little effect in our model.

To evaluate the effect of persistent homology analysis, we compare two various methods to build the snapshot group in our model (i.e., finding similar snapshots for a given snapshot). One is our persistent homology-based method (i.e., (6)), and the other is based on the cosine similarity metric (computing the similarity between two snapshots using the cosine similarity, i.e., no persistent homology analysis is done). As shown in Tables III and IV, we can see that our persistent homology-based solution outperforms the cosine similarity-based method, especially on the commercial dataset. For example, the MAP and NDCG@3 decreases by 2.4% and 3.2% using the cosine similarity-based solution, respectively. The reason could be that in the commercial dataset, there are very few users with similar click history, thus the

TABLE III
RESULTS OF ABLATION EXPERIMENTS ON THE AOL DATASET

	AOL Dataset													
	MAP		MRR		P@1		AveRank		NDCG@3		NDCG@5		NDCG@10	
PHPS	.7479	-	.7535	-	.6921	-	4.2233	-	.7348	-	.7513	-	.7738	-
Short-term Profile, Long-term Profile, and Group Profile														
wo. short	.7442	-0.5%	.7500	-0.5%	.6884	-0.5%	4.4464	-5.3%	.7308	-0.1%	.7481	-0.4%	.7692	-0.6%
wo. long	.7358	-1.6%	.7415	-1.6%	.6815	-1.5%	4.7108	-11.5%	.7213	-1.8%	.7387	-1.7%	.7586	-2.0%
wo. group	.6963	-6.9%	.7019	-6.8%	.6506	-6.0%	6.0490	-43.2%	.6776	-7.8%	.6923	-7.8%	.7137	-7.8%
Persistent Homology Analysis														
Cos-Similarity	.7428	-0.7%	.7485	-0.1%	.6840	-1.2 %	4.2796	-1.3%	.7295	-0.7%	.7470	-0.1%	.7690	-0.1%

TABLE IV
RESULTS OF ABLATION EXPERIMENTS ON THE COMMERCIAL DATASET

	Commercial Dataset													
	MAP		MRR		P@1		AveRank		NDCG@3		NDCG@5		NDCG@10	
PHPS	.4507	-	.4804	-	.2862	-	6.5810	-	.3990	-	.4663	-	.5407	-
Short-term Profile, Long-term Profile, and Group Profile														
wo. short	.4483	-0.5%	.4773	-0.6%	.2848	-0.5%	6.6932	-1.7%	.3970	-0.5%	.4627	-0.8%	.5382	-0.5%
wo. long	.4421	-1.9%	.4717	-1.8%	.2801	-2.1%	6.7463	-2.5%	.3884	-2.6%	.4550	-2.4%	.5319	-1.6%
wo. group	.4350	-3.5%	.4657	-3.1%	.2744	-4.1%	6.8630	-4.3%	.3827	-4.1%	.4489	-3.7%	.5263	-2.7%
Persistent Homology Analysis														
Cos-Similarity	.4399	-2.4%	.4693	-2.3%	.2776	-3.0%	6.7704	-2.9%	.3861	-3.2%	.4523	-3.0%	.5302	-1.9%

cosine similarity-based method is ineffective. Our persistent homology-based solution, however, can capture the topological features of user's click history which is often more robust compared to the cosine similarity-based method.

3) Improvements in Snapshot Establishment: When first using the snapshots to build groups, we found that some of the snapshots in the same group have few or none similar documents but still have high similarity when we using persistent homology analysis. After a closer inspection, the high similarity appears to be caused by the high-local similarity in the graph we build. Review the persistent homology analysis mentioned earlier, it mainly considers the relative position relationship between the points by calculating the distance between them. However, when the graph is big enough, there can be some similar subgraphs in different location of the graph. As Fig. 4(b) shows, these subgraphs may confuse the persistent homology analysis, making it hard to distinguish them and thus causing high similarities over those dissimilar snapshots.

To alleviate such situation, a conceivable method is adding some fixed markers into the snapshots, which is shown in Fig. 4(c). These markers can emphasize the absolute position of other points in the subgraph by calculating the distance between the markers and other points. Specifically, we randomly choose some of the documents in each graph and add them into the snapshots. These new snapshots can contain more information of the documents and the improvement is shown in Figs. 5 and 6. We can find the new snapshots have high degree of distinction, especially on the commercial dataset which have a bigger graph. For example, the MAP and NDCG@3 increase by 0.6% and 0.8% when using the new snapshots, respectively.

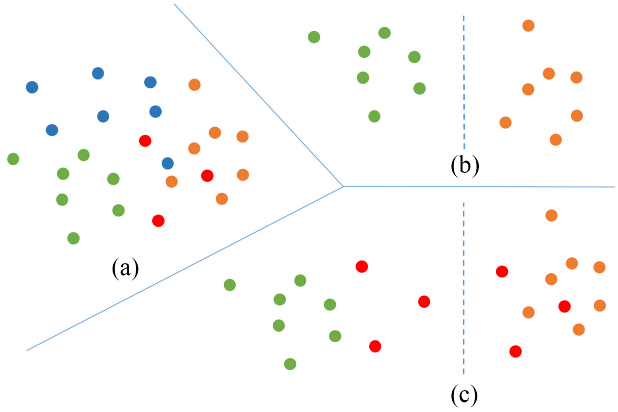


Fig. 4. Persistent homology analyze on graph (a) persistent homology analyze cannot distinguish the two subgraphs (green points and orange points) in (b) though they are at different positions on the graph. By adding markers (red points) into these two subgraphs as (c), the persistent homology can find the difference between them by calculating the distance between the points from markers and points in the subgraphs.

V. CONCLUSION AND DISCUSSION

In this article, we propose a new model, called PHPS, which encodes users' short-term, long-term and group-level profiles for personalized search. The novelty of our model is that we extract group-level features of users' search history by using a topological data analysis method, namely, persistent homology analysis. Such persistent-homology-based topological features

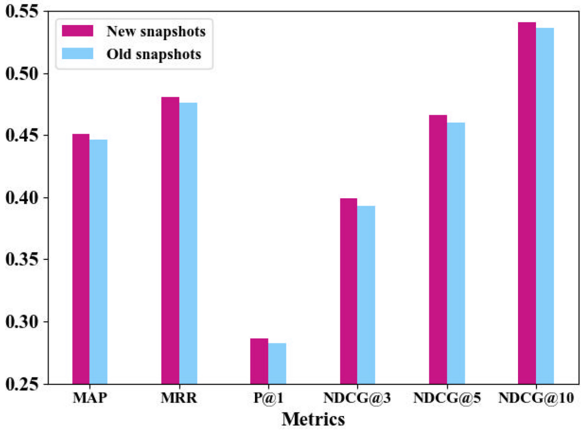


Fig. 5. Results of comparing the new snapshots and old snapshots on the commercial dataset.

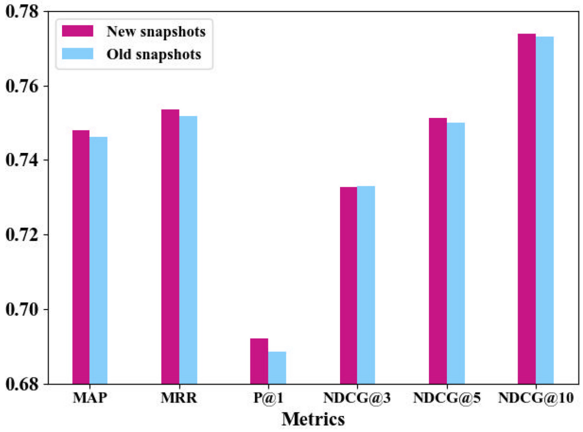


Fig. 6. Results of comparing the new snapshots and old snapshots on the AOL dataset.

are often robust with respect to noisy data and thus they are very useful for improving the performance of personalized search. We conduct extensive experiments to evaluate our model using two real-world datasets. The results show that our solution significantly outperforms the state-of-the-art models according to three commonly used precision metrics.

Based on the analysis conducted in this study, it is evident that the persistent homology-based method demonstrates remarkable performance in personalized search tasks, showcasing its robustness and effectiveness in constructing user's group profile, especially in the case with significant amount of noisy data. Moreover, there is still significant potential for further advancements in practical applications of topology analysis. Therefore, future research efforts should be directed toward the development and implementation of novel topological data analysis techniques that can facilitate more efficient and accurate retrieval of similar user. In addition, researchers can explore diverse models for user interest profile construction. One promising approach is aggregating more comprehensive topological information into the user profile extraction process, which has the potential to significantly augment the expressive capacity of the user interest profile. Such improvement in models' performance while preserving topological features can be the key to increased effectiveness of personalized search.

REFERENCES

[1] W. Ahmad, K.-W. Chang, and H. Wang, "Multitask learning for document ranking and query suggestion," in *Proc. ICLR*, 2012, pp. 1–12.

[2] A. Kashyap, R. Amini, and V. Hristidis, "SonetRank: Leveraging social networks to personalize search," in *Proc. CIKM*, 2012, pp. 2045–2049.

[3] P. N. Bennett, F. Radlinski, R. W. White, and E. Yilmaz, "Inferring and using location metadata to personalize Web search," in *Proc. SIGIR*, 2011, pp. 135–144.

[4] P. N. Bennett, K. Svore, and S. T. Dumais, "Classification-enhanced ranking," in *Proc. WWW*, 2010, pp. 111–120.

[5] P. N. Bennett et al., "Modeling the impact of short-and long-term behavior on search personalization," in *Proc. SIGIR*, 2012, pp. 185–194.

[6] C. J. C. Burges, K. M. Svore, Q. Wu, and J. Gao, "Ranking, Boosting, and model adaptation," Microsoft Res., Redmond, WA, USA, Rep. MSR-TR-2008-109, 2008.

[7] G. Carlsson, A. Zomorodian, A. Collins, and L. Guibas, "Persistence barcodes for shapes," *Int. J. Shape Model.*, vol. 11, pp. 149–188, Jan. 2005.

[8] M. J. Carman, F. Crestani, M. Harvey, and M. Baillie, "Towards query log based personalization using topic models," in *Proc. CIKM*, 2010, pp. 1849–1852.

[9] J. M. Chan, G. Carlsson, and R. Rabadan, "Topology of viral evolution," *Proc. Nat. Acad. Sci.*, vol. 110, no. 46, pp. 18566–18571, 2013.

[10] H. Choi et al., "Abnormal metabolic connectivity in the pilocarpine-induced epilepsy rat model: A multiscale network analysis based on persistent homology," *NeuroImage*, vol. 99, no. 8, pp. 226–236, 2014.

[11] M. K. Chung, P. Bubenik, and P. T. Kim, "Persistence diagrams of cortical surface data," in *Proc. Int. Conf. Inf. Process. Med. Imag.*, 2009, pp. 386–397.

[12] K. Collins-Thompson, P. N. Bennett, R. W. White, S. de la Chica, and D. Sontag, "Personalizing Web search results by reading level," in *Proc. CIKM*, 2011, pp. 403–412.

[13] D. Carmel et al. "Personalized social search based on the user's social network," in *Proc. CIKM*, 2009, pp. 1227–1236.

[14] D. Davis, G. Figueroa, and Y.-S. Chen, "SociRank: Identifying and Ranking Prevalent News Topics Using Social Media Factors," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 6, pp. 979–994, Jun. 2017.

[15] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.

[16] Z. Dou, R. Song, and J.-R. Wen, "A large-scale evaluation and analysis of personalized search strategies," in *Proc. WWW*, 2007, pp. 581–590.

[17] F. Zhang, W. Chen, M. Fu, F. Li, H. Qu, and Z. Yi, "An attention-based interactive learning-to-rank model for document retrieval," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 52, no. 9, pp. 5770–5782, Sep. 2022.

[18] S. Ge, Z. Dou, Z. Jiang, J.-Y. Nie, and J.-R. Wen, "Personalizing search results using hierarchical RNN with query-aware attention," in *Proc. CIKM*, 2018, pp. 347–356.

[19] M. Hajij, G. Zamzmi, and X. Cai, "Persistent homology and graphs representation learning," 2021, *arXiv:2102.12926*.

[20] M. Harvey, F. Crestani, and M. J. Carman, "Building user profiles from topic models for personalised search," in *Proc. CIKM*, 2013, pp. 2309–2314.

[21] J. Li, G. Liu, C. Yan, and C. Jiang, "Robust learning to rank based on portfolio theory and AMOSA algorithm," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 6, pp. 1007–1018, Jun. 2017.

[22] H. Lee, M. K. Chung, H. Kang, B.-N. Kim, and D. S. Lee, "Discriminative persistent homology of brain networks," in *Proc. 8th IEEE Int. Symp. Biomed. Imag.*, 2011, pp. 841–844.

[23] H. Lee, H. Kang, M. K. Chung, B.-N. Kim, and D. S. Lee, "Persistent brain network homology from the perspective of dendrogram," *IEEE Trans. Med. Imag.*, vol. 31, no. 12, pp. 2267–2277, Dec. 2012.

[24] S. Lu, Z. Dou, X. Jun, J.-Y. Nie, and J.-R. Wen, "PSGAN: A minimax game for personalized search with limited and noisy click data," in *Proc. SIGIR*, 2019, pp. 555–564.

[25] M. Bender et al., "Exploiting social relations for query expansion and result ranking," in *Proc. ICDE Workshops*, 2008, pp. 501–506.

[26] M. R. Morris and E. Horvitz, "SearchTogether: An interface for collaborative Web search," in *Proc. UIST*, 2007, pp. 3–12.

[27] M. Nicolau, A. J. Levine, and G. Carlsson, "Topology based data analysis identifies a subgroup of breast cancers with a unique mutational profile and excellent survival," *Proc. Nat. Acad. Sci.*, vol. 108, no. 17, pp. 7265–7270, 2011.

[28] D. Pachauri, C. Hinrichs, M. K. Chung, S. C. Johnson, and V. Singh, "Topology-based kernels with application to inference problems in Alzheimer's disease," *IEEE Trans. Med. Imag.*, vol. 30, no. 10, pp. 1760–1770, Oct. 2011.

[29] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using siamese BERT-networks," in *Proc. EMNLP-IJCNLP*, 2019, pp. 3982–3992.

[30] J. Reininghaus, S. Huber, U. Bauer, and R. Kwitt, "A stable multi-scale kernel for topological machine learning," in *Proc. CVPR*, 2015, pp. 1–8.

[31] R. Ghrist, "Barcodes: The persistent topology of data," *Bull. Am. Math. Soc.*, vol. 45, no. 1, pp. 61–75, 2008.

[32] S. Robertson and H. Zaragoza, "The probabilistic relevance framework: BM25 and beyond," *Found. Trends® Inf. Retrieval*, vol. 3, no. 4, pp. 333–389, 2009.

[33] A. Sieg, B. Mobasher, and R. Burke, "Web search personalization with ontological user profiles," in *Proc. CIKM*, 2007, pp. 525–534.

[34] G. Singh, F. Memoli, T. Ishkhanov, G. Sapiro, G. Carlsson, and D. L. Ringach, "Topological analysis of population activity in visual cortex," *J. Vis.*, vol. 8, no. 8, pp. 1–18, 2008.

[35] Y. Song, H. Wang, and X. He, "Adapting deep ranknet for personalized search," in *Proc. WSDM*, 2014, pp. 83–92.

[36] J. Teevan, S. T. Dumais, and E. Horvitz, "Potential for personalization," *ACM Trans. Comput.-Hum. Interact.*, vol. 17, no. 1, pp. 1–31, 2010.

[37] J. Teevan, M. R. Morris, and S. Bush, "Discovering and using groups to improve personalized search," in *Proc. WSDM*, 2009, pp. 15–24.

[38] T. A. Bjørklund, M. Götz, J. Gehrke, and N. Grimsmo, "Workload-aware indexing for keyword search in social networks," in *Proc. CIKM*, 2011, pp. 535–544.

[39] A. Vaswani et al., "Attention is all you need," in *Proc. NIPS*, 2017, pp. 1–11.

[40] V. D. Silva and R. Ghrist, "Coverage in sensor networks via persistent homology," *Algebra. Geom. Topol.*, vol. 7, no. 1, pp. 339–358, 2007.

[41] T. Vu, D. Q. Nguyen, M. Johnson, D. Song, and A. Willis, "Search Personalization with embeddings," in *Advances in Information Retrieval*. Cham, Switzerland: Springer Int. Publ., 2017, pp. 598–604.

[42] T. T. Vu, D. Song, A. Willis, S. N. Tran, and J. Li, "Improving search personalisation with dynamic group formation," in *Proc. SIGIR*, 2014, pp. 951–954.

[43] H. Wang, X. He, M.-W. Chang, Y. Song, R. W. White, and W. Chu, "Personalized ranking model adaptation for Web search," in *Proc. SIGIR*, 2013, pp. 323–332.

[44] R. V. D. Weygaert et al., "Alpha, betti and the megaparsec universe: On the topology of the cosmic Web," in *Transactions on Computational Science XIV: Special Issue on Voronoi Diagrams and Delaunay Triangulation*. Berlin, Germany: Springer, 2011, pp. 60–101.

[45] P. Wu et al., "Optimal topological cycles and their application in cardiac trabeculae restoration," in *Proc. Inf. Process. Med. Imag.*, 2017, pp. 80–92.

[46] J. Yao, Z. Dou, and J. Wen, "Employing personal word embeddings for personalized search," in *Proc. SIGIR*, 2020, pp. 1359–1368.

[47] Y. Zhou, Z. Dou, B. Wei, R. Xie, and J.-R. Wen, "Group based personalized search by integrating search behaviour and friend network," in *Proc. SIGIR*, 2021, pp. 92–101.

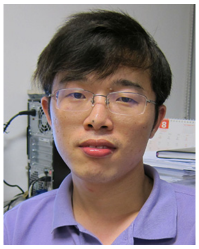
[48] Y. Zhou, Z. Dou, and J.-R. Wen, "Encoding history with context-aware representation learning for personalized search," in *Proc. SIGIR*, 2020, pp. 1111–1120.

[49] Y. Zhou, Z. Dou, and J.-R. Wen, "Enhancing re-finding behavior with external memories for personalized search," in *Proc. WSDM*, 2020, pp. 789–797.

[50] Y. Zhou, Z. Dou, Y. Zhu, and J.-R. Wen, "PSSL: Self-supervised learning for personalized search with contrastive sampling," in *Proc. CIKM*, 2021, pp. 2749–2758.

Yuchen Meng received the B.S. degree in computer science and technology from the Beijing Institute of Technology, Beijing, China, in 2022, where he is currently pursuing the M.S. degree in computer science and technology.

His research interests include big data management, graph generation, and topological data analysis.



Rong-Hua Li received the Ph.D. degree in computer science from The Chinese University of Hong Kong, Hong Kong, in 2013.

He is currently a Professor with the Beijing Institute of Technology, Beijing, China. His research interests include graph data management and mining, social network analysis, graph computation systems, and graph-based machine learning.

Hongchao Qin received the B.S. degree in mathematics and the M.E. and Ph.D. degrees in computer science from Northeastern University, Shenyang, China, in 2013, 2015, and 2020, respectively.

He is currently a Postdoctoral Fellow with the Beijing Institute of Technology, Beijing, China. His current research interests include social network analysis and data-driven graph mining.

Xiang Wu received the B.S. degree in computer science and technology from the Beijing Institute of Technology, Beijing, China, in 2021, where he is currently pursuing the M.S. degree with the Department of Computer Science.

His research interests include graph neural networks, graph representation learning, and data mining.

Huanzhong Duan received the M.S. degree in computer science from the Beijing University of Posts and Telecommunications, Beijing, China.

He is currently an Expert Researcher with the Wechat Search Application Department, Tencent, Shenzhen, China. His research interests include machine learning and data mining.

Yanxiong Lu received the M.S. degree in pattern recognition and intelligent system from Xidian University, Xi'an, China.

He is currently an Expert Researcher with the Wechat Search Application Department, Tencent, Shenzhen, China. He has published several papers on some conferences, such as ACL and AAAI. His research interests include information retrieval, natural language processing, and mechine learning.

Guoren Wang received the B.Sc., M.Sc., and Ph.D. degrees in computer science from the Department of Computer Science, Northeastern University, Shenyang, China, in 1988, 1991, and 1996, respectively.

He is currently a Professor with the Department of Computer Science, Beijing Institute of Technology, Beijing, China. He has published more than 100 research papers. His research interests include XML data management, query processing and optimization, bioinformatics, high-dimensional indexing, parallel database systems, and cloud data management.